Overview of Swirlds Hashgraph

Leemon Baird
baird@swirlds.com
May 31, 2016

The hashgraph data structure and Swirlds consensus algorithm provide a new platform for distributed consensus. This paper gives an overview of some of its properties, and comparisons with the Bitcoin blockchain. In this paper, the term “blockchain” will generally refer to the system used in Bitcoin, rather than the large number of variants that have been proposed.

The goal of a distributed consensus algorithm is to allow a community of users to come to an agreement on the order in which some of them generated ***transactions***, when no single member is trusted by everyone. In this way, it is a system for generating trust, when individuals do not already trust each other. The Swirlds hashgraph system achieves this along with being ***fair***, ***fast***, ***provable***, ***Byzantine***, ***ACID compliant***, ***efficient, inexpensive, timestamped, DoS resistant***, and optionally ***non-permissioned***. This is what those terms mean:

The hashgraph is ***fair***, because no individual can manipulate the order of the transactions. For example, imagine a stock market, where Alice and Bob both try to buy the last available share of a stock at the same moment for the same price. In blockchain, a miner might put both those transactions in a single block, and have complete freedom to choose what order they occur. Or the miner might choose to only include Alice’s transaction, and delay Bob’s to some future block. In the hashgraph, there is no way for an individual to affect the consensus order of those transactions. The best Alice can do is to invest in a better internet connection so that her transaction reaches everyone before Bob’s. That’s the fair way to compete. Alice won’t be able to bribe the miner to give her an unfair advantage, because there’s no single person responsible for the order.

The hashgraph is also ***fair*** in another way, because no individual can stop a transaction from entering the system, or even delay it very much. In block chain, a transaction can be delayed by one or two mining periods, if many of the miners are refusing to include it. In alternatives to blockchain based on leaders, this delay can be extremely long, until the next change of leader. But in the hashgraph, attackers cannot stop a member from recording a transaction in any way other than cutting off their internet access.

The hashgraph is ***fast***. It is limited only by the bandwidth. So if each member has enough bandwidth to download 4,000 transactions per second, then that is how many the system can handle. That would likely require only a few megabits per second, which is a typical home broadband connection. And it would be fast enough to handle all of the transactions of the entire Visa card network, worldwide. The Bitcoin limit of 7 transactions per second can clearly be improved in various ways. Though some ways of improving it, such as a gigantic block size, could actually make the fairness of the system even worse.

The hashgraph is ***provable***. Once an event occurs, within a couple of minutes everyone in the community will know where it should be placed in history. More importantly, everyone will know that everyone else knows this. At that point, they can just incorporate the effects of the transaction, and then discard it. So in a minimal crypto currency system, each member (each "full node" in blockchain terminology) needs only to store the current balance of each wallet that isn't empty. They don't need to remember any old blocks. They don't need to remember any old transactions. That shrinks the amount of storage from Bitcoin's current 60 GB to a fraction of a single gigabyte. That would even fit on a typical smartphone.

The hashgraph is ***Byzantine***. This is a technical term meaning that no single member (or small group of members) can prevent the community from reaching a consensus. Nor can they change the consensus once it has been reached. And each member will eventually reach a point where they know for sure that they have reached consensus. Blockchain does not have a guarantee of Byzantine agreement, because a member never reaches certainty that agreement has been achieved (there's just a probability that rises over time). Blockchain is also non-Byzantine because it doesn't automatically deal with network partitions. If a group of miners is isolated from the rest of the internet, that can allow multiple chains to grow, which conflict with each other on the order of transactions. It is worth noting that the term "Byzantine" is sometimes used in a weaker sense. But here, it is used in its original, stronger sense that (1) every member eventually knows consensus has been reached (2) attackers may collude and (3) attackers even control the internet itself (with some limits). Hashgraph is Byzantine, even by this stronger definition.

The hashgraph is ***ACID compliant***. This is a database term, and applies to the hashgraph when it is used as a distributed database. A community of members uses it to reach a consensus on the order in which transactions occurred. After reaching consensus, each member feeds those transactions to that member's local copy of the database, sending in each one in the consensus order. If the local database has all the standard properties of a database (ACID: Atomicity, Consistency, Isolation, Durability), then the community as a whole can be said to have a single, distributed database with those same properties. In blockchain, there is never a moment when you know that consensus has been reached. But if we were to consider 6 confirmations as achieving "certainty", then it would be ACID complaint in the same sense as hashgraph.

The hashgraph is 100% ***efficient***, as that term is used in the blockchain community. In blockchain, work is sometimes wasted mining a block that later is considered stale and is discarded by the community. In hashgraph, the equivalent of a "block" never becomes stale.

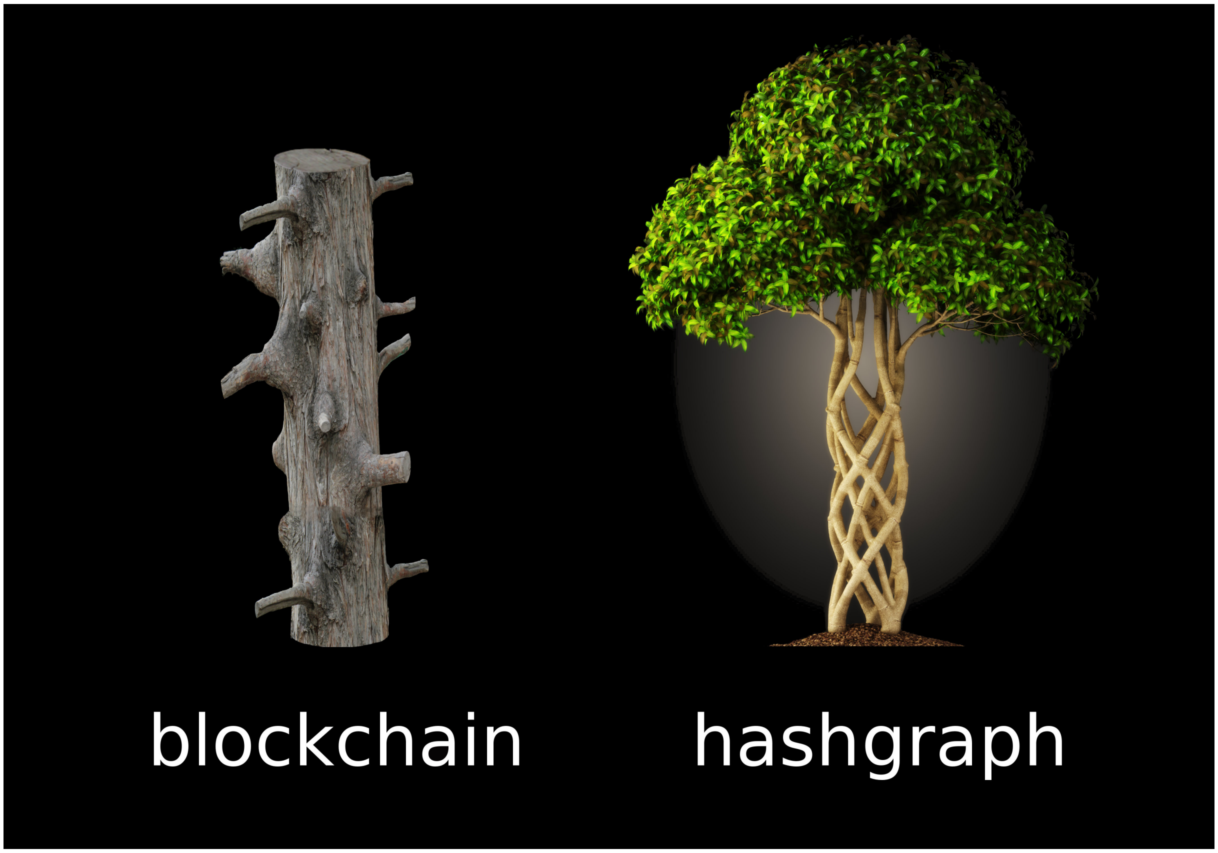
The hashgraph is ***inexpensive***, in the sense of avoiding ***proof-of-work***. In Bitcoin, the community must waste time on calculations that slow down how fast the blocks are mined. As computers become faster, they'll have to do more calculations, to keep the rate slow. The calculations don't have any useful purpose, except to slow down the community. This requires the serious miners to buy expensive, custom hardware, so they can do this work faster than their competitors. But hashgraph is 100% efficient, no matter how fast its "blocks" are mined. So it doesn't need to waste computations to slow itself down. (Note: there are blockchain variants that also don't use proof-of-work; but Bitcoin does require proof-of-work).

The hashgraph is ***timestamped***. Every transaction is assigned a consensus time, which is the median of the times at which each member first received it. This is part of the consensus, and so has all the guarantees of being Byzantine and provable. If a majority of the participating members are honest and have reliable clocks on their computer, then the timestamp itself will be honest and reliable, because it is generated by an honest and reliable member, or falls between two times that were generated by honest and reliable members. This consensus timestamping is useful for things such as smart contracts, because there will be a consensus on whether an event happened by a deadline, and the timestamp is resistant to manipulation by an attacker. In blockchain, each block contains a timestamp, but it reflects only a single clock: the one on the computer of the miner who mined that block.

The hashgraph is ***DoS resistant***. Both blockchain and hashgraph are distributed in a way that resists Denial of Service (DoS) attacks. An attacker might flood one member or miner with packets, to temporarily disconnect them from the internet. But the community as a whole will continue to operate normally. An attack on the system as a whole would require flooding a large fraction of the members with packets, which is more difficult. There have been a number of proposed alternatives to blockchain based on *leaders* or *round robin*. These have been proposed to avoid the proof-of-work costs of blockchain. But they have the drawback of being sensitive to DoS attacks. If the attacker attacks the current leader, and switches to attacking the new leader as soon as one is chosen, then the attacker can freeze the entire system, while still attacking only one computer at a time. Hashgraph avoids this problem, while still not needing proof-of-work.

The hashgraph is optionally ***non-permissioned***, while still avoiding the cost of proof-of-work. A *permissioned* system is one where only trusted members can participate. An *open* system is not permissioned, and allows anyone to participate. Standard blockchain can be open if it uses proof-of-work, but variants such as proof-of-stake typically have to be permissioned in order to be secure. A hashgraph system can be designed to work in a number of different ways. One of the more interesting is to use proof-of-stake, allowing members to vote proportional to their ownership of a particular cryptocurrency. A good cryptocurrency might be widely used, so that it is difficult for an attacker to corner the market by owning a large fraction of the entire money supply. If a large fraction of the currency owners all participate in a hashgraph system, then proof-of-stake will make it safe from *Sybil attacks*, which are attacks by hordes of sock-puppet fake accounts. Such a system would be secure even if it were not permissioned, while still avoiding the cost of proof-of-work.

The following figure illustrates why hashgraph has these desirable properties.



Why does hashgraph have these properties? Because it’s like a tree that’s braided, not pruned.

In both blockchain and hashgraph, any member can create a transaction, which will eventually be put into a container (the “block”), and will then spread throughout the community. In blockchain, those containers are intended to form a single, long chain. If two miners create two blocks at the same time, the community will eventually choose one to continue, and discard the other one. It’s like a growing tree that is constantly having all but one of its branches chopped off.

In hashgraph, every container is used, and none are discarded. So all the branches continue to exist forever, and eventually grow back together into a single whole. This is more efficient. Furthermore, blockchain fails if the new containers arrive too quickly, because new branches are sprouting faster than they can be pruned. That is why blockchain needs proof-of-work or some other mechanism to artificially slow down the growth. But in hashgraph, nothing is thrown away. So there is no harm in the structure growing quickly. Every member can create transactions and containers whenever they want. So it is very simple, and tends to be very fast.

Finally, because the hashgraph doesn’t require pruning, it is simpler, which allows more powerful mathematical guarantees, such as *Byzantine agreement* and *fairness*. Distributed databases such as Paxos are Byzantine, but not fair. Blockchain is neither Byzantine nor fair. But the Swirlds hashgraph is both Byzantine and fair.