DRAFT

The DCS Theorem

Greg Slepak

October 4, 2017

Abstract. We present a probability proof of the *DCS Triangle* [1][2]. We use the triangle to show decentralized consensus systems cannot scale to support the transactional demands of centralized consensus systems, and therefore any *single system* can have *Decentralization*, *Consensus*, or *Scale*, but not all three properties simultaneously.

Definitions

A *system* is defined as any set of components (see Scope) following *precise rules* in order to provide service(s) to the users of the system. These services constitute the system's *intended behavior*.

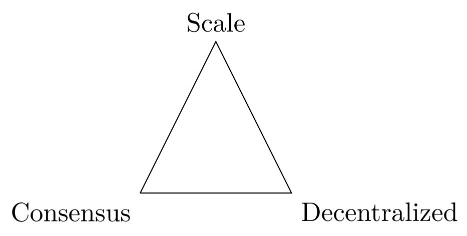
In other words, a system S consists of a set of components, called its *scope* $\{S\}$, and a program ("state transition function", f_S), that together define the system's *intended behavior*, which means: upon receipt of message m, S uses f_S to update the internal state from s to s' and send back reply y within a time interval S_τ.

$$S(t) = \begin{cases} \{S\} & = \{component_1, component_2, \cdots\} \\ f_S(m, s) & = \{s', y\} \end{cases}$$

We note, additionally:

- The *scope* $\{S\}$ may change over time, but there are always several components of a consistent *type* (i.e. "all systems always have at least one *CPU*, one *developer*, and one *user*")
- The system's state s includes all data necessary for the system to compute f_S given a message m
- S is considered *compromised* if it fails to perform its intended behavior within the interval S_τ

We will proceed to prove that any single such system may possess, at most, two of three properties:



- **Consensus** means the system's state, s is a *shared state* that is updated by nodes running a *consensus algorithm* over a network, and that furthermore, the output of *consensus algorithm* determines the network's accepted output of f_S, and whether or not f_S completes within S_τ.
- **Scale** means the system is capable of handling the transactional demands of any competing system providing the same service to the same arbitrary set of users across the globe (*"at scale"*).[1]
- **Decentralized** means the system has no *single point of failure or control* (SPoF). Another way to state this is: the system continues to perform its intended behavior if any single element is removed from $\{S\}$, and no single component in $\{S\}$ has the power to redefine f_S on its own.

Systems whose intended behavior can be modified without the consent of their users are considered *centralized* due to the presence of a central point of control over the definition of the system.

Decentralization Scope & Relativity of Decentralization

Implicit in our definition of a *decentralized system* is the idea that the system is not compromised. A non-functioning system does not fulfill its intended behavior, and therefore, by our definition, is not decentralized.

Imagine a decentralized system S, whose intended behavior (its purpose) is to maintain the integrity of a database while being responsive to queries. It does so by attempting to eliminate all single points of failure within a given *scope.*

Definition. The *scope* of a system refers to all subcomponents and all entities "reasonably relevant" to a system's functioning.

If we consider the scope of our "decentralized" database to be a computer with two CPUs and two hard disks (one primary, another backup), then we can say S is "decentralized" at $t = 0$ (has no single point of failure). However, if at $t = 1$ one of the hard disk fails, it is no longer decentralized since now there does exist a single component capable of compromising the entire system.

[1]Examples of "services" include: streaming video, sending messages, maintaining balances on a ledger, etc.

This means:

- Whether or not a system is decentralized can change over time.
- Any system can be called "decentralized" if we define the scope narrowly enough.
- All decentralized systems can be called "centralized" if we define their scope broadly enough.[2]

The narrowing and enlarging of the scope is called the *relativity of decentralization*, and it is why first agreeing on a reasonable definition for a system's scope is vital before deciding whether or not it is "decentralized".

Computational throughput of consensus systems

The *computational throughput* $T(S)$ of any consensus system depends on three factors:

1. The computational power of each *consensus participant* (those able to select transactions which are *written* to the shared state).[3]
2. The amount of time the consensus algorithm considers messages to be lost (the *timeout* period).
3. The consensus *threshold* the consensus algorithm uses to decide whether consensus has been reached (e.g. "how big of a quorum is required").

Note that if the computational power of a consensus participant is significantly less than that of the other participants, they are more likely to be excluded from the deciding quorum for several reasons:

- If there are no network partitions to determine otherwise, fast consensus participants will process messages more quickly and therefore will be first to create a quorum.
- If there are enough fast consensus participants to create a large enough quorum to exceed the system's consensus threshold, then there is no need to wait for the slow participants to move the system forward.
- Slow consensus participants are more likely than fast consensus participants to hit the system's timeout period for processing and responding to messages, and therefore are more at risk of being excluded from the consensus process entirely.

Therefore, $T(S)$ is a function that is *limited by the slowest consensus participants not excluded in the deciding quorum.*

[2]The entire Internet could be considered centralized if we include the entire solar system as part of the scope. The "single point of failure" could be the Earth itself, its atmosphere, the Sun, etc. Or, perhaps in the not distant future, a single ISP.

[3]Note that participants who are only allowed to *verify* state, but not participate in *creating* it, are not considered consensus *participants*. Instead, they are called state *validators*.

Coordination costs

Relevant for our proof is the notion of *coordination costs*, or the difficulty for one entity to engage another and work toward a common goal.

For example, when Bitcoin was first launched, it would be difficult for any miner to find enough collaborating miners to create a cartel with 51%+ of the hash power, simply because there were many "relevant miners" (consensus participants) distributed all over the world.

Today, however, there are significantly fewer consensus participants in Bitcoin, and it is much easier to (1) identify them, and (2) bring them together in a single room to coordinate around some goal. Therefore, we say the coordination costs are lower today than before.

We can approximate the coordination costs $C(S)$ of any consensus system as simply the number of consensus participants:

$$C(S) = \texttt{num\_consensus\_participants}(\{S\})$$

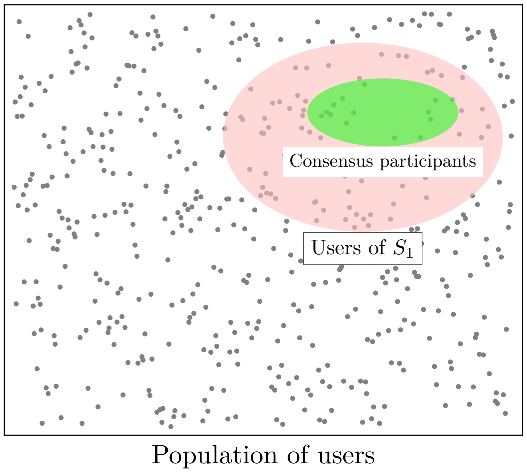


Fig. 1: TODO: explain / elaborate on this figure and perhaps show system dynamic.

Proof

Theorem 1. *A consensus system that is decentralized initially, tends toward centralization as it scales to meet the demands of competing (and functionally equivalent) centralized consensus systems.*

We begin with the following axioms accepted as true:

Axiom 1. *In any sufficiently large population (at scale), individual access to computational power is not distributed uniformly. Most individuals have access to average computational power, and a few have access to large amounts.*

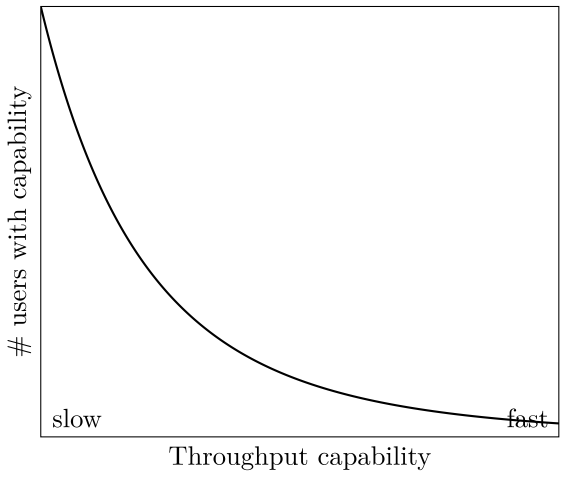


Fig. 2: Visualization of (Axiom 1).

Axiom 2. *In any two systems offering the same service to the same large population, the transactional demands of the average user converge at scale.*

From those axioms, we derive the following lemmas:

Lemma 1. *Let S_1 and S_2 be consensus systems offering the same service to the same population of users at scale. Let N be a function returning the number of users of a given system. If $N(S_1) > N(S_2)$, and if both systems remain uncompromised as they gain users, then $T(S_1) > T(S_2)$ converges to a true statement as the value $N(S_1) - N(S_2)$ increases.*

Proof. This follows directly from (Axiom 2) and from the assertion that both systems remain uncompromised (indicating they are indeed processing messages on time as they gain more users). □

Lemma 2. *Let S be a decentralized consensus system with a growing number of users. The number of users who can act as consensus participants decreases, and therefore, $C(S)$, decreases at scale.*

Proof. As a system S gains more users the throughput of the system, $T(S)$ increases, per (Lemma 1). By (Axiom 1), the number of individuals capable of handling that throughput decreases. Since S starts out decentralized, there does not exist a single entity controlling who the participants are, and therefore some users who previously chose to play the role consensus participants are no longer able to process messages fast enough to participate in consensus. Therefore $C(S)$ decreases as well. □

Lemma 3. *Let S be a decentralized consensus system. Coordination costs for S decrease at scale.*

Proof. For our proof we are only interested in what happens to decentralized consensus systems at scale. By (Lemma 2) and our definition of $C(S)$, it directly follows that the coordination costs of decentralized consensus systems decrease at scale because there are fewer consensus participants, which makes it easier for consensus participants to identify each other. $\square$

Lemma 4. *Let S be a decentralized consensus system. The probability that $\{S\}$ contains a cartel capable of colluding to censor transactions increases at scale, and therefore S tends toward centralization.*

Proof of the Main Theorem. Per (Lemma 3), fewer consensus participants means the probability of a colluding group (cartel) increases. If a cartel coordinates to successfully compromise consensus, the system is no longer decentralized (per our definitions of consensus and decentralized). If it is easier for consensus participants to identify each other, it follows that it is easier for anyone to identify a quorum of consensus participants. Either a successful attempt at coercing this quorum, or a successful attempt at collusion would represent system failure. In decentralized systems, system failure indicates there was a single point of failure (see example in Scope). Therefore, the probability of centralization increases. $\square$

Acknowledgements

The author would like to thank Anya Petrova for her assistance with the paper and feedback on the proof.

References

[1] T. McConaghy, "The DCS Triangle," 10-Jul-2016. [Online]. Available: https://medium.com/the-bigchaindb-blog/the-dcs-triangle-5ce0e9e0f1dc.

[2] G. Slepak, "Slepak's Triangle," *Rebooting Web-of-Trust*, 17-Oct-2016. [Online]. Available: https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2016/blob/master/topics-and-advance-readings/Slepaks-Triangle.pdf.