02 Protokoły i algorytmy rozproszonego konsensusu

Yang Xiao[1], *Ning Zhang*[2], *Jin Li*[3], *Wenjing Lou*[1] *i Y. Thomas Hou*[1]

1 Virginia Polytechnic Institute and State University, Blacksburg, VA, USA
2 Washington University in St. Louis, St. Louis, MO, USA
3 Guangzhou University, Guangzhou, Chiny

2.1. Wprowadzenie

Algorytmy odpornego na awarie konsensusu (*fault-tolerant consensus*) zostały poddane wnikliwym badaniom w kontekście systemów rozproszonych. Decydując o przekazywaniu informacji w sieci rozproszonych procesorów, algorytmy te umożliwiają procesorom uzgodnienie stanu danych i podejmowanie przez nie tych samych działań w odpowiedzi na żądania wykonania usług, niezależnie od tego, czy wszystkie komponenty działają poprawnie i czy wszystkie kanały komunikacyjne poprawnie przekazują dane. Gwarancja osiągnięcia konsensusu ma kluczowe znaczenie dla poprawnego działania systemów rozproszonych.

Blockchain, będący również systemem rozproszonym, wymaga protokołu konsensusu umożliwiającego wszystkim węzłom sieci uzgodnienie jednego łańcucha historii transakcji, z założeniem możliwości nieprzewidzianego zachowania ewentualnych uszkodzonych węzłów lub węzłów złośliwych. W chwili pisania tego tekstu w świecie kryptowalut funkcjonowało ponad tysiąc różnych inicjatyw i projektów korzystających z ponad 10 klas protokołów konsensusu. W tym rozdziale omówimy podstawy

klasycznego konsensusu odpornego na awarie i jego zastosowań w przetwarzaniu rozproszonym oraz przedstawimy kilka popularnych protokołów konsensusu wykorzystywanych w rozwiązaniach opartych na blockchainie.

Rozdział ten ma następującą strukturę: podrozdział 2.2 wprowadza podstawy odpornego na awarie konsensusu w systemie rozproszonym oraz dwa praktyczne protokoły konsensusu dla przetwarzania rozproszonego. Podrozdział 2.3 przedstawia protokół konsensusu Nakamoto – pionierski protokół konsensusu oparty na dowodzie pracy (PoW – Proof of Work), wykorzystany po raz pierwszy w sieci Bitcoin. Podrozdział 2.4 przedstawia kilka rozwijanych aktualnie blockchainowych protokołów konsensusu, nieopartych na PoW oraz ich scenariusze zastosowań. Podrozdział 2.5 przedstawia ocenę jakościową i porównanie wyżej wymienionych protokołów konsensusu. Podrozdział 2.6 kończy rozdział i podsumowuje filozofię protokołów konsensusu dla rozwiązań opartych na blockchainie.

2.2. Odporny na awarie konsensus w systemie rozproszonym

Wszystkie komponenty systemu rozproszonego dążą do osiągnięcia wspólnego celu, mimo swojej geograficznej separacji. Konsensus, mówiąc najprościej, oznacza, że komponenty systemu osiągają porozumienie w sprawie jakichś wartości lub danych. W rzeczywistych systemach komponenty i ich kanały komunikacyjne są podatne na nieprzewidziane awarie i działania przeciwników. W tym punkcie omówimy problem konsensusu systemów wykorzystujących komunikaty[1], dla dwóch typów awarii ich komponentów – *załamania* (*crash failure*) i dla awarii bizantyjskiej (*Byzantine failure*). Następnie zajmiemy się dwoma praktycznymi algorytmami konsensusu, które tolerują takie awarie komponentów w systemach rozproszonych. Terminów procesor, węzeł i komponent będziemy dla wygody używać w tym punkcie wymiennie.

2.2.1. Model systemu

Istnieją trzy główne czynniki konsensusu w systemie rozproszonym: synchronizacja sieci, awarie komponentów i protokół konsensusu.

[1] Istnieje inny typ systemów rozproszonych: system z pamięcią dzieloną (*shared-memory system*). Więcej szczegółów można znaleźć w publikacji [1]. W tym rozdziale skoncentrujemy się na systemie przekazującym komunikaty, ze względu na jego podobieństwo do łańcucha bloków.