ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY
A MAGYAR TUDOMÁNY ÜNNEPE
Az MTA programsorozata
Mérnöki Szimpózium a Bánkiban
(ESB 2025)

Konszenzus algoritmusok összehasonlító elemzése pénzügyi blockchain alkalmazásokban

Comparative Analysis of Consensus Algorithms in Financial Blockchain Applications

[1]Forman Norbert, [2]Szathmári Gergő

[1]*Budapesti Gazdaságtudományi Egyetem, Budapest, Magyarország, forman.norbert@uni-bge.hu,*

[2]*OTP Bank Nyrt., Budapest, Magyarország, Gergo.Szathmari@otpbank.hu*

Összefoglalás

A pénzügyi szektor digitalizációjában az elosztott főkönyvi technológiák működésének kulcselemei a konszenzus algoritmusok. Jelen tanulmányunkban a banki környezetben releváns konszenzus mechanizmusok – különösen a Raft, Paxos és DAG-alapú megoldások – összehasonlító elemzését mutatjuk be. A vizsgálat egy magyar kereskedelmi bank saját fejlesztésű, nagy teljesítményű, Raft-alapú blockchain rendszerének bemutatására épül, amely rendszer egy K+F projekt eredményeként került kifejlesztésre. Eredményeink szerint az architekturális optimalizációk mellett a Raft-alapú megoldások akár nagyságrendekkel felülmúlhatják a nyilvános blockchain rendszerek teljesítményét.

Kulcsszavak: konszenzus algoritmusok, Raft, Paxos, DAG, blockchain technológia, elosztott rendszerek, teljesítmény összehasonlítás

Abstract

During the digitalization of the financial sector, consensus algorithms are a key component of distributed ledger technology operations. This paper provides a comparative analysis of consensus mechanisms relevant to banking environments, with a particular focus on Raft, Paxos, and DAG-based approaches. The study presents a proprietary, high-performance Raft-based blockchain system developed by a Hungarian commercial bank as the outcome of a research and development (R&D) project. Our results indicate that, with appropriate architectural optimizations, Raft-based solutions can outperform public blockchain systems by one or more orders of magnitude.

Keywords: consensus algorithms, Raft, Paxos, DAG, blockchain technology, distributed systems, performance comparison.

1 Bevezetés

A kereskedelmi bankok – mint pénzügyi szolgáltató intézmények – informatikai rendszerei egyszerre kényszerülnek óriási mennyiségű tranzakciók kezelésére (alacsony késleltetés, determinisztikus működés), ugyanakkor a szigorú auditálhatóság biztosítására. Az elosztott főkönyvi technológiák (Distributed Ledger Technologies - DLT) különösen relevánsak a naplózás, integritásbiztosítás és egyéb megfelelőségi kritériumok biztosítása szempontjából, mivel a rekordok kriptográfiai összekapcsolásával, utólagos módosítás esetén az inkonzisztencia azonnal

detektálható. A DLT-rendszerek gyakorlati alkalmazhatóságát döntően a konszenzusmechanizmus határozza meg, a résztvevő csomópontoknak azonos állapotot kell jelezniük részleges hibák és változó hálózati késleltetések mellett is [1].

A nyilvános, ún. engedélymentes (nincs központi jogosultságkezelés a részvételhez) blockchain-ekben – mint például a Bitcoin által alkalmazott megoldás – elterjedt konszenzusmechanizmusok (pl. Proof of Work - PoW) ugyan erős hibatűréssel rendelkeznek, ugyanakkor jelentős energiafelhasználással többletköltséget is okoznak, továbbá korlátozott skálázhatósággal rendelkeznek. Banki, nagyvállalati terhelés mellett ezért tipikusan a zárt, ún. engedélyes topológia a célszerű kialakítás, ahol a résztvevők (hálózati csomópontok) köre ismert és kontrollált, továbbá a hibatűrési modell gyakran a leállási (crash) hibák kezelésére szűkíthető [2].

Jelen közlemény a Raft, Paxos és DAG-alapú (Directed Acyclic Graph – irányított körmentes gráf) megoldások banki kontextusú összehasonlítását mutatja be, alapvetően egy, az OTP Bank Nyrt-ben megvalósított K+F projekt valós eredményeire alapozva, amely egy nagy teljesítményű, Raft-alapú, engedélyes blockchain megoldásra épült. Az eredmények adott tudományterülethez való hozzájárulása kettős: (i) döntéstámogató értékelési dimenziók formalizálása a konszenzusmechanizmus kiválasztásához; (ii) a 10^5 TPS (másodpercenkénti tranzakció) nagyságrend eléréséhez szükséges architekturális és üzemeltetési feltételek optimalizációjának bemutatása.

2 Elméleti háttér és kapcsolódó munkák

2.1 CAP (Consistency Availability Partition tolerance) szemlélet

A konszenzusprobléma klasszikus megfelelési kritériumai a terminálás (a megfelelő csomópontok hoznak döntést), a megállapodás (nem születhet két eltérő döntés), valamint az érvényesség (a döntés a bemenetekkel konzisztens) [1]. A banki DLT-tervezésben a hibatűrési modell megválasztása meghatározza a protokoll komplexitását és teljesítményét: CFT (Crash Fault Tolerance – leállási hibatűrés) esetén a csomópontok legfeljebb leállnak, míg BFT (Byzantine Fault Tolerance – bizánci hibatűrés) esetén rosszindulatú viselkedés is feltételezhető. A CAP szemléletében a banki rendszerek tipikusan a konzisztenciát és a partíciótűrést preferálják, továbbá hálózati partíció esetén kontrolláltan korlátozhatják csak az elérhetőséget [3].

2.2 Raft és Paxos (CFT) mechanizmusok

A Raft algoritmus célja, hogy a konszenzus elérése és implementálása egyszerűbb legyen, mint a Paxos esetén: ún. leader–follower architektúrát alkalmaz és explicit választási, illetve naplóreplikációs szabályokat rögzít [2]. A Paxos matematikailag elegáns, ugyanakkor implementációs komplexitása viszonylag magas, a gyakorlatban a Multi-Paxos változat a jellemző, ahol stabil leader (kiemelt, privilegizált csomópont) mellett kedvező throughput érhető el, leader-váltáskor viszont erőforrás igényesebb, így költségesebb az új állapotfelépítés és az újraszinkronizáció [4].

2.3 DAG-alapú megoldások és finalitás

A DAG-alapú főkönyvi struktúrák a lánc helyett ún. irányított, körmentes gráfot (nincs olyan út, amely visszavezetne ugyanabba a csomópontba) használnak, így a párhuzamosan létrejövő bejegyzések könnyen integrálhatók a struktúrába, csökkentve a versengésből eredő „elveszett teljesítmény” jelenségét. A megközelítés előnyei a kiterjedt párhuzamosság és a kedvező throughput, ugyanakkor a konfliktuskezelés jellemzően komplexebb, továbbá banki környezetben,

dokumentált üzemeltetési tapasztalat viszonylag korlátozotton ismerhető jelenleg [5].

3 Kutatási módszertan

Az elemzés az alábbi módszertan szerint kerül bemutatásra: (i) szakirodalmi áttekintést adunk a konszenzusmechanizmusok megfelelőségi és teljesítmény tulajdonságairól; (ii) technikai dokumentáció elemzést és architekturális leírást az OTP Bank Nyrt. engedélyes blockchain típusú fejlesztéséről; (iii) előző két pontra alapozva, összehasonlító értékelést adunk a jellemző banki követelmények mentén (TPS, késleltetés, finalitás, hibatűrés, auditálhatóság). A teljesítményértékek értelmezésénél külön kezeljük a laboratóriumi benchmarkokat és az éles üzemi környezetben várható end-to-end teljesítményt, mivel utóbbiban az integrációs és kontroll-folyamatok (authN/authZ, kulcskezelés, monitorozás) is késleltetést eredményeznek.

4 Esettanulmány: nagy teljesítményű, Raft-alapú banki engedélyes típusú blockchain megoldás

4.1 Üzleti célok és alkalmazási kontextus

Az esettanulmány tárgya az OTP Bank Nyrt. belső banki környezetében tervezett ún. engedélyes blockchain megoldás, amelynek elsődleges feladata, nagy volumenű eseménynaplók és tranzakció rekordok megváltoztathatatlanságának biztosítása. A projekt egyik legfontosabb üzleti motivációja a megfelelőségi (compliance), illetve auditfolyamatok gyorsítása, továbbá azok támogatása volt: a hagyományos központi log-tárolásban a visszamenőleges módosítás kockázatát, csak komplex jogosultság- és ellenőrzési rendszerekkel lehet ellenőrizni, illetve mérsékelni, míg a hash-láncolt, replikált napló esetén az utólagos módosítás kriptográfiailag detektálható, a bizonyítás pedig reprodukálható.

A megoldás tipikus felhasználási területei közé tartozik a kritikus banki tranzakciók és automatizált folyamatok (pl. RPA) bemenő/kimenő állományainak integritásvédelme, illetve olyan eseményláncok rögzítése, amelyek egy későbbi ellenőrzés során alátámasztják a döntések és feldolgozások időrendjét. Ezekből fakadóan a követelményrendszer: (i) magas írási áteresztőképesség (cél: legalább 100 000 TPS), (ii) alacsony és kis szórású késleltetés, (iii) magas rendelkezésre állás csomópont-kiesés esetén, (iv) determinisztikus viselkedés és visszakövethető audit nyom, (v) vállalati hálózati és kriptográfiai standardokkal történő integráció (TLS, ipari hash).

4.2 Miért a Raft? – hibatűrési modell és auditálhatóság

A konszenzusmechanizmus kiválasztásánál a legfontosabb döntési pont a hibatűrési modell meghatározása. A belső, banki engedélyes blockchain hálózatban a csomópontok üzemeltetése központilag kontrollált, így a rosszindulatú (Byzantine) viselkedés valószínűsége tipikusan alacsonyabb, mint nyilvános hálózatokban. Ennek megfelelően a projekt megvalósításának megtervezése a CFT-modell szerint történt: a rendszernek csomópont-leállást, hálózati késleltetést és átmeneti elérhetetlenséget tolerálnia kellett, de nem kellett alapértelmezetten kezelnie a protokollt szándékosan megszegő résztvevőket.

A Raft választása több okból is indokolt volt: (i) egyszerű, auditálható állapotgép-modell; (ii) leader-follower naplóreplikáció alacsony kommunikációs overheaddel; (iii) gyors leader-választás és helyreállás; (iv) determinisztikus commit-szabály (többségi replikáció után commit), amely banki környezetben jól illeszkedik a tranzakcióvéglegességi elvárásokhoz [2].

4.3 Architektúra: hálózat, szerializáció és adattárolás

A rendszer architektúrája négy rétegre bontható: konszenzus (Raft), hálózati protokoll, adattárolás és kriptográfia. Hálózati szinten TCP biztosítja a megbízható üzenetküldést, opcionális TLS titkosítással. A csomópontok közötti üzenetek rögzített sémák szerint kerültek szerializálásra, azaz adatstruktúrák determinisztikus bájtsorozattá alakítására, amely biztosította az elosztott rendszer csomópontjai közötti egységes értelmezést és feldolgozást. A hatékony és optimalizált bináris kódolás csökkentette a CPU- és sávszélesség igényt, ami közvetlenül növelte a leader oldali feldolgozható klienskérések számát.

A teljesítmény kritikus eleme az írásra optimalizált adattárolás. A tárolási modell szétválasztja a metaadatokat (mutatók, hossz, offset, hash-lánc elemek) és a tényleges adatfolyamot. A metaadat-stream atomikusan írható, így a commitolt rekordokra vonatkozó hivatkozások konzisztensen rögzíthetők, miközben a nagy mennyiségű adatok folyamatosan, append-only jelleggel kerültek elhelyezésre. Ezzel a kialakítással az auditálás lényeges része – a hash-ek és metaadatok ellenőrzése – gyorsan elvégezhető anélkül, hogy minden rekordot teljes egészében vissza kellene olvasni!

4.4 Teljesítményoptimalizálás: tipikus szűk keresztmetszetek

A 100 000 TPS nagyságrend elérését a teljes rendszer paraméterei együttesen határozzák meg, nem csupán a konszenzus protokoll. Tipikus szűk keresztmetszet a leader csomópont CPU terhelése (szerializáció, hash-számítás, napló-írás), illetve az I/O-alrendszer tartós írási sebessége. A hálózati késleltetés a véglegesítéshez szükséges késleltetést (commit-latency-t) közvetlenül befolyásolja, mivel a leadernek többségi visszaigazolásra van szüksége. A kriptográfiai műveletek (hash, aláírás, TLS) költsége nagy TPS mellett kumulatív módon jelentkezik, ezért gyakori stratégia a bejegyzések batchelésének (összecsomagolás) és pipelining-jának alkalmazása, amit mi is alkalmaztunk a projekt során.

Az üzenet érdemi, üzleti adattartalmának (payload) mérete és a kliens kérések mintázata szintén fontos paraméterek. Kis payloadok-nál a protokoll és a rendszerhívások többletterhelése (overhead) dominál, nagy payloadok-nál a hálózat és az adattár válik limitáló tényezővé. Banki naplózási use case-ekben bevált megközelítés a payloadok hash-elt lenyomatának főkönyvön belül végrehajtott (on-chain) rögzítése és a tényleges adatok kontrollált, külső tárolása, így a DLT-réteg a bizonyítási lánc szerepét tölti be, miközben az érzékeny adatok hozzáférése külön jogosultsági szint szerint kezelhető.

4.5 Integráció, adat-életciklus és governance

A banki alkalmazásokhoz az engedélyes blockchain réteg tipikusan szolgáltatásként kapcsolódik (API gateway, üzenetsor, eseménybusz). Az integrációs minta meghatározza a valós késleltetést: a konszenzuson túl megjelenik a kliensoldali tranzakció-előkészítés, a jogosultság-ellenőrzés, valamint a naplóhoz kapcsolt metaadatok előállítása. Az audit egyik alapvető igénye, a rekord-életciklus visszakereshetősége, ezért célszerű standard metaadat-sémát bevezetni (folyamat-azonosító, rendszer-azonosító, verzió, időbélyeg, hash-ujjlenyomat), amely akár későbbi megfelelőségi (compliance) jelentések és incidens-analitikák alapját képezhetik.

A CFT-alapú konszenzus akkor tekinthető megfelelőnek, ha az irányítási keret (governance) biztosítja a résztvevő csomópontok konfigurációs integritását és azonos verzió készletét. Gyakorlati kontrollok: konfiguráció-aláírás, immutable infrastruktúra (IaC), fokozatos, kísérleti jellegű bevezetés, valamint folyamatos integritás ellenőrzés mind a node-ok konfigurációján, mind binárisain. A kulcskezelés (aláírókulcsok, TLS tanúsítványok) HSM-mel támogatható, és a kulcs-rotációt a változáskezelési folyamatokba kell ágyazni.

4.6 Döntési dimenziók összefoglalása

Az 1. táblázat a banki környezetben releváns döntési dimenziók mentén foglalja össze a vizsgált megközelítéseket.

1. táblázat: Konszenzus algoritmusok összehasonlítása banki környezetben [2][4][5]

Mechanizmus	Tranzakció véglegessége / hibatűrés	Banki alkalmazhatóság
Raft (CFT)	Szabályvezérelt commit, crash hibák	Magas throughput, auditálható; engedélyes
Multi-Paxos	stabil leader mellett hatékony	Robusztus, de komplex implementáció
DAG-alapú	Párhuzamosítás, tranzakció véglegessége komplex	Ígéretes, de korlátozott banki tapasztalat jelenleg

5 Összehasonlító elemzés

5.1 Teljesítmény (TPS és késleltetés)

Raft esetén a tranzakció akkor tekinthető véglegesnek, ha a bejegyzés többségi replikációja megtörtént; alacsony késleltetésű topológiában ez tipikusan milliszekundumos nagyságrendű. Multi-Paxos stabil leader mellett hasonló throughputot nyújthat, de a protokoll és a helyes implementáció komplexebb, a leader-váltás pedig gyakran hosszabb konvergenciaidőt okoz [4]. A DAG-megközelítések a párhuzamos bejegyzéseket jobban hasznosíthatják, azonban a véglegesítés és konfliktuskezelés auditálása és formalizálása banki környezetben több kockázatot hordozhat, különösen heterogén topológia és változó kliens-terhelés esetén [5].

5.2 Biztonság, auditálhatóság és megfelelőség

A banki megfelelőség szempontjából a determinisztikus működés és a visszakereshető auditnyom elsőrendű. Engedélyes topológiában a CFT-modell gyakran elégséges, feltéve, hogy az üzemeltetési kontrollok erősek, és a résztvevői kör kezelhető. Több intézményre történő kiterjesztés esetén megfontolandó BFT vagy a federal trust modell, illetve a governance és felelősségi körök formalizálása (csomópont-jogosultságok, frissítési rend, incidenskezelés).

5.3 Döntéstámogató kiválasztási keretrendszer

A konszenzus mechanizmus kiválasztásához javasolt dimenziók: TPS potenciál, késleltetés tranzakció véglegesség (finalitás) követelmény (determinisztiku vs. probabilisztikus), auditálhatóság és üzemeltetési komplexitás. A 2. táblázatban szereplő Raft tipikusan akkor optimális, ha a cél magas throughput és alacsony késleltetés zárt környezetben; Multi-Paxos akkor, ha meglévő Paxos-alapú komponensekhez kell illeszkedni; DAG-megközelítések pedig akkor, ha a párhuzamosítás stratégiai követelmény és a szervezet képes a komplexebb finalitási modell és governance kezelésére.

5.4 Összehasonlító értékelési mátrix

A 2. táblázat kvalitatív módon összegzi a vizsgált megközelítéseket banki követelmények mentén.

2. táblázat: Konszenzus mechanizmusok kvalitatív összevetése banki környezetben

Mechanizmus	TPS potenciál	Késleltetés	Tranzakció véglegessége	Auditálhatóság	Üzemeltetési komplexitás
Raft (CFT)	magas	alacsony	szabály vezérelt	magas	közepes
Multi-Paxos	magas	alacsony-közepes	szabály vezérelt	közepes	magas
DAG-alapú	magas	alacsony	komplex	közepes	közepes-magas

Megjegyzés: a besorolások tipikus implementációk alapján értendők; a tényleges értékeket a topológia, node-szám, payload és üzemeltetési kontrollok jelentősen befolyásolják.

6 Következtetések

A tanulmány rámutatott, hogy a banki környezet követelményei (auditálhatóság, determinisztikus finalitás, magas throughput) jellemzően az engedélyes, CFT-alapú konszenzusmechanizmusok felé terelik a tervezést. Az esettanulmány alapján a Raft pragmatikus és hatékony választás volt: egyszerűen auditálható, stabil leader mellett magas teljesítményt nyújt, és rendszer-szintű optimalizációkkal elérhető a 10^5 TPS nagyságrendű cél. Jövőbeli irány a WAN-topológiák kvantitatív vizsgálata (késleltetés, quorum elhelyezés), valamint a kiterjesztéshez szükséges irányítási keretrendszer formalizálása.

Köszönetnyilvánítás

Jelen publikáció a 2020-1.1.2-PIACI-KFI-2020-00061 számú *„Következő generációs Banki megoldások kifejlesztése”* című projekt keretében az NKFIH támogatásával jött létre.

7 Hivatkozások

[1] Lamport, L. (1998). The part-time parliament. ACM Transactions on Computer Systems, 16(2), 133–169. https://doi.org/10.1145/279227.279229

[2] Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm (Raft). USENIX ATC 2014, 305–319.

[3] Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News, 33(2), 51–59. https://doi.org/10.1145/564585.564601

[4] Lamport, L. (2001). Paxos made simple. ACM SIGACT News, 32(4), 51–58.

[5] Baird, L. (2016). The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance. Technical Report.