

Dr. Szabó Zsolt bírálói kérdéseire adott válaszok

MTA doktora értekezés szerzője, címe:

Kertész Attila

Komplex, elosztott rendszerek viselkedésének elemzése

Először is szeretném megköszönni Dr. Szabó Zsoltnak, hogy időt és energiát szánt MTA doktori értekezésem alapos átolvasására, tanulmányozására, és megfogalmazta opponensi véleményét. Köszönöm az értekezéssel kapcsolatos hasznos észrevételeit és kérdéseit, amelyeket az alábbiakban válaszolok meg:

"1. A doktori műben foglalt eredmények jelentős részét a kérelmező másokkal együttműködve érte el. A benyújtott doktori mű és a tézispontok megfogalmazása alapján abban a saját teljesítményének aránya kétséget kizáróan nem állapítható meg. Ezért kérem a tézisek új megfogalmazását, különös tekintettel a 2.1, 2.2, 2.3, 3.1, 3.2, 3.3, 4.1, 4.2, és 4.4 tézispontokra, egyértelműsítve a közös teljesítményből a kérelmező részesedését."

Válasz: Mind a négy téziscsoportban az elért eredmények nagy részét hazai vagy nemzetközi csapatmunka keretében végeztem. A 2-4. téziscsoportokban ismertetett problémákon témafelelős és témavezető kutatóként dolgoztam, így az én hozzájárulásom a pontos kutatási célok definiálása és az elméleti modellek kidolgozása volt. Az általam vezetett és felügyelt doktori és egyetemi hallgatókkal közösen végzett feladatok az irodalomkutatás és a tesztelési, validációs feladatok elvégzése volt. Az implementációs feladatokat főként a következő doktori hallgatóim végezték: Pflanzner Tamás, Márkus András, Baniata Hamza, és Mishra Biswajeeban.

Az egy-két mondatos altézis kimondásokban a tömörségre fókuszáltam, így itt valóban nem részleteztem minden esetben az egyes kontribúciókat, de az azokat követő bekezdésekben véleményem szerint a tudományterületemen elfogadott gyakorlatnak megfelelően, kellően megfogalmaztam a hozzájárulásokat, amelyek egyértelműsítik részesedésem. Számszerű, százalékos részesedés megfogalmazása a szűkebb kutatási területemen nem szokás, viszont megerősítem, hogy a kapcsolódó publikációkhoz társszerzőimmal egyenlő arányban járultunk hozzá.

A fentiek alapján a következő módon pontosítom és egészítem ki a nevezett altézis kimondásokat, az azokat követő részletezés alapján:

2.1. Meghatároztam az MQTT protokollt alkalmazó üzenetbrókerek csoportosítási kategóriáit, és Mishra Biswajeebannal közösen kiértékeltek és osztályoztuk az MQTT protokollt alkalmazó üzenetbrókereket.

2.2. Megterveztem egy módszert IoT-Felhő rendszerek részleges szimulációs vizsgálatára. A módszer alapján közösen kidolgoztuk a MobIoTSim IoT eszközszimulátort, melynek implementációját Pflanzner Tamás végezte. A szimulátor képes valós szenzor adatokat generálni és küldeni valós hálózati környezetben felhős szolgáltatásoknak.

2.3. Megterveztem egy általános felhős átjáró (gateway) architektúrát. A módszer alapján közösen kidolgoztunk egy átjáró szolgáltatást, melynek implementációját Pflanzner Tamás végezte. A

megoldás képes a MobIoTSim-mel és/vagy valós IoT eszközökkel generált adatok hatékony kezelésére és vizualizálására. Vass Imrével és SZBK-s kollégáival közösen megterveztük az IoLT Okos Cserép megfizethető fenotipizálási platformot, melyhez Pflanzner Tamással közösen kidolgoztuk a módszer specializált változatát, az IoLT átjáró alkalmazást, mely képes üvegházi növények távoli monitorozására.

3.1. Meghatároztam az IoT-Köd-Felhő rendszerek szimulációs megoldásainak csoportosítási kategóriáit, és Márkus Andrással közösen megterveztünk egy taxonómiát, mellyel osztályoztuk és elemeztük a létező szimulátorokat. Az osztályozási vizsgálatokhoz egy új, szoftver metrikákon alapuló elemzési módszert terveztem meg, melynek implementációját Márkus András végezte.

3.2. Megtervezttem egy bővíthető modellt IoT eszközök, kód erőforrások és szolgáltatói árazási sémák leírására. A sémák alkalmazásával Márkus Andrással közösen kidolgoztuk a modellt megvalósító DISSECT-CF-IoT, majd DISSECT-CF-Fog szimulátorokat, és kiértékeljük azokat különféle használati eseteket megvalósító scenáriókkal. A szimulátorok és a használati esetek implementációját Márkus András végezte.

3.3. Megtervezttem egy metodológiát IoT-Köd-Felhő alkalmazások viselkedéselemzéséhez. Ezt felhasználva Márkus Andrással közösen kidolgoztunk különféle erőforrás allokációs stratégiákat, melyek algoritmusai képesek csökkenteni az alkalmazások futási idejét és a felhasznált erőforrások költségét, egyúttal optimalizálva az IoT-Köd-Felhő infrastruktúra kihasználtságát. Az algoritmusokat közösen kiértékeljük a DISSECT-CF-Fog szimulátorban, és bemutattuk hatékonyságukat. A használati esetek implementációját és futtatását Márkus András végezte.

4.1. Megtervezttem egy módszert kód és blokklánc rendszerek integrálására, melyek vizsgálatára Baniata Hamzával közösen kidolgoztunk egy nyílt, bővíthető szimulátort, a FoBSim-et. A szimulátor alkalmazhatóságát számos blokklánc-telepítési scenárió elemzésével szemléltettük. A szimulátor és a konszenzusos algoritmusok implementációját Baniata Hamza végezte.

4.2. Baniata Hamzával közösen kidolgoztunk egy módszert a blokklánc hálózatok teljesítményének és konzisztenciájának elemzésére, melyet kiértékelünk a FoBSim eszközzel. Kidolgoztunk és kiértékelünk egy adattovábbítást optimalizáló protokollt, mely dinamikus szomszédválasztással teszi lehetővé a szomszédok számának minimalizálását. A kapcsolódó algoritmusok implementációját Baniata Hamza végezte.

4.4. Baniata Hamzával közösen kidolgoztunk egy metodológiát magánszféra-védelmet megvalósító kód-alapú blokklánc rendszerek tervezésére. Módszereit Baniata Hamzával közösen kiértékelünk egy kód és blokklánc infrastruktúrával támogatott felhős feladatütemezési alkalmazás szimulációjával, valamint egy valós intézmény akkreditációs és tanúsítvány validációs alkalmazással. Az algoritmusok, a szimulációs környezet és az alkalmazás implementációját Baniata Hamza végezte.

"2. A doktori mű általában nem tartalmazza a kidolgozott/alkalmazott algoritmus leírását, hanem részletek helyett a jelölt publikációit hivatkozza. Az algoritmusok bemutatása megkönnyítené az értekezés szimulációinak részletesebb megértését."

Válasz: Való igaz, hogy területi korlátok miatt az értekezésben törekedtem a főbb eredmények tömör bemutatására, így az egyes megoldások részletes implementációját és algoritmusait nem tudtam kellő részletességgel szemléltetni. Emellett minden ilyen "tömörítésnél" feltüntettem a

kapcsolódó publikációk hivatkozásait, amelyek tartalmazzák a szakterület elvárásait kielégítő gondos és részletes algoritmusok, kiértékelések és validációk leírásait, magyarázatait. Mivel az első tézis csoport tartalmazza a legszerteágazóbb kutatásokat, itt volt szükség a legtömörebb megfogalmazásra.

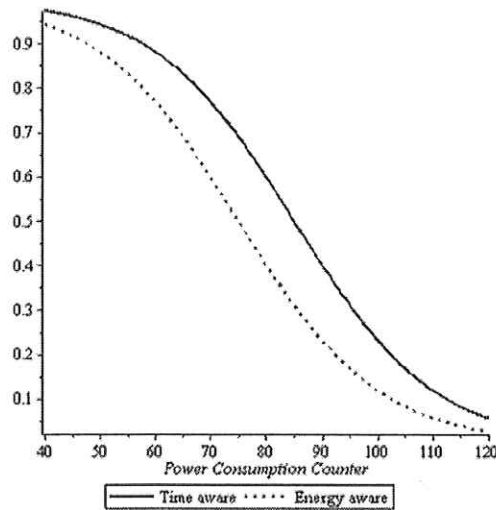
"3. Hogyan történt a felhő föderációk adatközpont vizsgálatára kidolgozott Pliant algoritmus ütemezési együtthatójának a megválasztása? Milyen a speciális szigmoid függvény, és milyen normalizálás történt?"

Válasz: A felhős adatközpontokban elérhető csomópontokat (fizikai hosztokat) a CloudSim szimulátorban történő modellezés során három paraméterrel jellemeztük ([9] a kapcsolódó publikáció az értekezésben): CPU terheltségi szint (PUC - power usage counter, százalék), áram fogyasztási érték (PCC - power consumption counter, Watt) és a processzorok száma (PROC - number of processors). Attól függően, hogy végrehajtási időre vagy energiafogyasztásra minimalizáltunk a Pliant algoritmussal, az alábbi szigmoid függvény következő paraméterezését használtuk, hogy az egyes jellemzőket normalizáljuk a [0-1] intervallumba:

$$f_{\alpha,\lambda}(x) = \frac{1}{1 + e^{\lambda(x-\alpha)}}$$

Property	Time		Energy	
	Alpha	Lambda	Alpha	Lambda
PUC	0.5	-4.0	0.5	-4.0
PCC	85.0	-0.08	75.0	-0.08
PROC	1.0	0.8	1.0	0.8

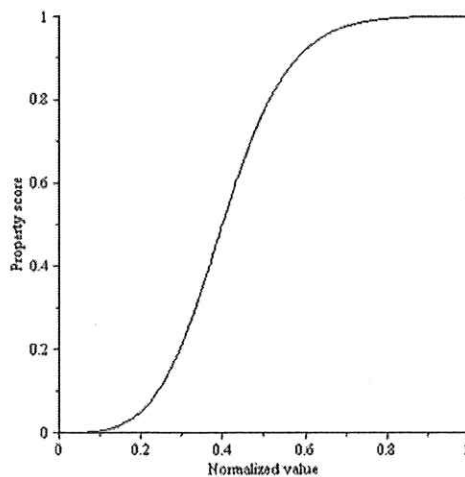
A jellemzőkre vonatkozóan, minél közelebb van az adott érték 1-hez, annál jobbnak számít a döntés során. Ezeket a paramétereket előzetes szimulációs mérésekkel állítottuk be az alkalmazott konkrét infrastruktúrához. A λ (Lambda) paraméter megválasztása a futtató infrastruktúra statikus tulajdonságaitól függ, az α (Alpha) paraméter pedig az adott tulajdonság értékének hasznosságát állítja be. A következő ábra szemlélteti a két stratégia (azaz a minimalizálás célja) közti különbséget:



Ezt követően a kapott értékeket az adott jellemző fontossága alapján skáláztuk a következő Kappa függvénnyel:

$$\kappa_{\nu}^{\lambda}(x) = \frac{1}{1 + \left(\frac{\nu}{1-\nu} \frac{1-x}{x}\right)^{\lambda}}$$

A következőképpen néz ki a Kappa függvény az alkalmazott $\nu=0.4$ (score középértéke) és $\lambda=3.0$ (meredekség) paraméterekkel:



Ezt követően minden virtuális gépre (VM) aggregáltuk az egyes jellemzőkre számított értékeket. Az így kapott súlyok alapján választottuk ki a végrehajtandó feladathoz (alkalmazás komponenshez) a futtató VM-et.

A normalizációs lépésekben alkalmazott paramétereket a jellemzők általunk megítélt fontossága alapján, valamint előzetes szimulációs mérésekkel állítottuk be. Ezt a Pliant algoritmust a későbbiekben a DISSECT-CF-Fog szimulátorban is alkalmaztuk többféle használati eset modellezésére és végrehajtási stratégia kiértékelésére. Legutóbbi kutatásunkban a múltbeli

jellemzőértékek gépi tanulással történő előrejelzését is felhasználtuk az aggregált súlyok meghatározásához, amely módszer az első lépés a fent statikusan meghatározott paraméterek automatizált kiválasztásához:

- A. Márkus, V. D. Hegedűs, J. D. Dombi, A. Kertész: Synergizing Fuzzy-based Task Offloading with Machine Learning-driven Forecasting for IoT. In Proc. of 8th IEEE International Conference on Fog and Edge Computing (ICFEC'24), Philadelphia, USA, 06-09 May, 2024.

"4. Miért szükséges a Service Level Agreement (SLA) leírására egy új nyelvet bevezetni? Valamelyik jelenleg is használt jelölő nyelv (markup language), mint az XML, miért nem megfelelő?"

Válasz: A szolgáltatások területén a szolgáltatás-minőséget illetően számos faktor alapján jellemezhetjük az egyes szolgáltatásokat. Ilyen tulajdonságok többek között a rendelkezésre állás folyamatossága, a válaszidő, vagy az igénybevétel ára. Attól függően, hogy milyen kutatási vagy üzleti szolgáltatási területet vizsgálunk, eltérő céllal tarthatjuk nyilván ezeket a tulajdonságokat és azok állapotát, pl. rendszer monitorozás, árazási információk dokumentálása, automatizált szolgáltatás választás, vagy elvárt szolgáltatási szintek szerződéses rögzítése. Ezeket az információkat tetszőleges formátumú dokumentumokban rögzíthetnénk, akár XML-ben is. Viszont amint egy egyedi céllal történő használaton túl, egy közösség által merül fel az igény bizonyos szolgáltatás-minőségi tulajdonságok nyilvántartására, szabványosítási problémával szembesülünk. Az adott közösségnek egy sémát kell kidolgoznia, melyben megegyeznek az egyes paraméterek megnevezésén és értéktartományán. Ez is kifejezhető egy XML sémával. Pontosan ezt az utat járta be az Open Grid Forum (OGF) szervezet a WS-Agreement¹ standard kidolgozásával a web szolgáltatás alapú grid rendszerek területén, amelyet hosszú évek munkájával fejlesztettek ki. A WS-Trust² szintén egy XML-alapú nyelv lett, melyet az Organization for the Advancement of Structured Information Standards (OASIS) szervezet hozott létre, biztonságos üzenetküldések megvalósítására. Az értekezésben [106]-os hivatkozással ellátott, az S-CUBE EU FP7 kiválósági hálózatban számos kutató együttműködésével készült cikkben ezen példákkal együtt 18 SLA nyelvet neveztünk meg, hasonlítottunk össze és vettünk alapul egy SLA meta-modell kidolgozásához. Bár ezek nagy része XML-alapú, létrehozásukat az adott alkalmazási területen dolgozó közösség szükségesnek érezte az egységes alkalmazhatóság végett. Az általunk végzett kutatás abban segít, hogy rendszerezett formában megismerhessük ezen nyelvekben alkalmazott metrikákat, összehasonlíthassuk az egyes megoldásokat, és adott esetben megfelelő nyelvet válasszunk a szolgáltatás-minőség jellemzésére egy számunkra alkalmazandó területen.

"5. Manapság milyen gyakorisággal (frekvenciával) továbbíthatók szenzormérések adatai (pl. 256 KB) az IoT-Fog-Cloud rendszerek esetén? Hogyan módosul ezeknek a rendszereknek a megbízhatósága az adattovábbítás gyakoriságának a függvényében? Mennyi a felhasználói oldal átlagos vételi ideje?"

Válasz: Az IoT-Köd-Felhő rendszerek esetén a szenzorok által generált adatok továbbítási gyakoriságát általában a szenzorok mintavételezési frekvenciája határozza meg. Ez az intervallum alkalmazásfüggő és igen széles skálán mozoghat: az egyszerűbb hőmérséklet és páratartalom szenzorok (pl. DHT-22) esetén a mintavételezési frekvencia igen lassú, mindössze 0.5 Hz, ami 2

1 <https://ogf.org/documents/GFD.192.pdf>

2 <https://specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf>

másodpercenkénti mérést jelent, de egy általános célú gázérzékelő esetén is a mintavételezési frekvencia tipikusan 10 Hz, a következő viszonylag friss publikáció alapján:

- Saoutieff E, Polichetti T, Jouanet L, Faucon A, Vidal A, Pereira A, Boisseau S, Ernst T, Miglietta ML, Alfano B, et al. A Wearable Low-Power Sensing Platform for Environmental and Health Monitoring: The Convergence Project. *Sensors*. 21(5):1802, 2021. <https://doi.org/10.3390/s21051802>

Ez az érték tovább emelkedhet bizonyos alkalmazások esetében (a kHz-es tartományba), de ez túlmutat a hagyományos értelemben vett IoT használati eseteken, és inkább az elektronikai rendszerekben játszik nagyobb szerepet.

A gyakorlatban az adatok továbbítási gyakorisága az IoT alkalmazástól függ, amely tipikusan másodpercenkénti, percenkénti vagy óránkénti felbontással történik. Például egy egyszerű hőmérséklet szenzor egy okosotthonban elhelyezve nem feltétlenül igényel gyakori adatátvitelt, míg egy forgalomfigyelő kamera sokkal sűrűbb adatátvitelt eredményez a gyorsan változó környezet miatt.

A gyakoribb adattovábbítás, különösen nagy kiterjedésű szenzorhálózat esetén, a szolgáltatás minőségének romlásához vezethet, mivel több adatot kell átmozgatni és feldolgozni a hálózaton. Ebben az esetben könnyen kialakulhat egy hálózati vagy számítási csomóponton az úgynevezett szűk keresztmetszet (bottleneck-effect), amely a hirtelen fellépő, nagy mennyiségű adat okozta túlterhelést jelenti. Ez a jelenség hatással lehet a felhasználói oldal vételi idejére is, hiszen a gyakoribb adattovábbítás generálta hálózati terhelés és a nagyobb adatmennyiség feldolgozása növelheti az alkalmazás késleltetését is. Az ebből a hatásból eredő problémákat egyrészt a tervezés során lehet figyelembe venni, és az üzenetovábbító alrendszer (pl. MQTT bróker) az alkalmazás igényeinek megfelelően felkonfigurálni. Ilyen jellegű skálázhatósági kísérleteket az értekezésben [129]-es hivatkozású munkánkban végeztünk. Amennyiben nem tudjuk előre megbecsülni az átlagos terheltséget, célszerű skálázódást támogató üzenetbróker megvalósítást választanunk.

A felhasználói oldal vételi idején az IoT eszközök által generált adatok szerver oldali eltárolását vagy feldolgozását érthetjük. Erre nem tudunk átlagos időt mondani. Egyrészt alkalmazásfüggő ez az idő, tehát a választott adatbázis mentési idejétől függ, másrészt a korábban említett üzenetbróker típusától és beállításától, valamint a használt hálózati infrastruktúra paramétereitől is függ (pl. sáv szélesség, távolság). Ezen összetevők közül a [129]-es hivatkozású cikkünkben a különböző típusú üzenetbrókerek késleltetését vizsgáltuk, mely eredmények egy részét az értekezés 2.2-es fejezetében is bemutattam.

"6. Hasonlítsa össze a vizsgált blokklánc megoldásokat a DAG (directed acyclic graph) technológiával. IoT eszközöket tartalmazó rendszerek esetén a javasolt szomszédválasztó protokoll helyett nem lenne előnyösebb a DAG alkalmazása?"

Válasz: A DAG definíció szerint egy gráf típus, mely irányított élekkel rendelkezik és körmentes, azaz az irányított éleken keresztül bejárva nem juthatunk el ugyanabba a csúcsba. A blokklánc rendszerek esetében megkülönböztetjük az adat réteget a hálózati rétegtől, az előbbiben az adatok tárolása, az utóbbiban a blokklánc csomópontok közötti kommunikáció zajlik.

A hálózati réteget tekintve, a blokklánc hálózat, melyben a csomópontokon futó bányászok egymással kommunikálnak, jellemzően egy peer-to-peer hálózat, melyben a bányászokhoz több

szomszédos bányász (peer) kapcsolódik véletlenszerű kiválasztással (pl. a Bitcoinban egy teljes bányászcsomópontra akár 99 szomszéd is juthat). Jellemzően a szabványos pletyka protokollt használják az adatoknak a hálózaton belüli továbbítására, ami azt jelenti, hogy minden egyes bányásznak adatokat kell küldenie minden egyes szomszédnak, amelyhez csatlakozik. Amennyiben ezt a hálózatot gráfként ábrázoljuk, az nem fog DAG tulajdonságokkal rendelkezni. Ha megpróbálnánk a szomszédságot és a kommunikációs irányokat ennek megfelelően kialakítani, azaz egy DAG-alapú struktúrát kialakítani, az így létrejövő rendszer súlyos biztonsági sebezhetőséggel rendelkezne, mivel nehezen lehetne biztosítani az egy kézben lévő rosszindulatú bányászok megfelelő arányát (ami kevesebb kell legyen, mint 51%). A szomszédok számának csökkentése viszont kedvező hatással lehet a kommunikációs késleltetés csökkentésére. Pontosan ez volt az a kutatási probléma, amelynek megoldására a DONS protokollt javasoltuk (mely az értekezésben a [31]-es hivatkozású cikk, és eredményeit a 4.2.3-as alfejezetben foglaltam össze). A DONS protokollban tett javaslatunkban a blokklánc rendszer peer-to-peer hálózati struktúrájából egy minimális feszítőfát alakítunk ki, hogy a bányászok valóban a legkevesebb szomszéddal kommunikáljanak, ahelyett, hogy a szabványos pletyka protokollban az összes szomszédnak továbbítanák adataikat, mégis megtartva a biztonsági előírásokat.

Az adat rétegben a blokklánc csomópontok (azaz a bányászok) által helyben vezetett főkönyv lehet lineáris blokklánc (minden szülő blokk pontosan egy gyermek blokkal van összekötve), vagy DAG-szerű blokklánc (minden szülő blokk több gyermek blokkal is össze lehet kötve). A tradicionális blokklánc rendszerek, mint pl. a Bitcoin és az Ethereum a lineáris adatstruktúrát (láncot) alkalmazza, míg újabb megoldások, mint pl. az IOTA, DAG-alapú adattárolást alkalmaznak. A következő cikkben olvashatunk egy részletes összehasonlítást a két adattárolási módszer között:

- Qin Wang, Jiangshan Yu, Shiping Chen, and Yang Xiang. SoK: DAG-based Blockchain Systems. ACM Comput. Surv. 55, 12, Article 261, 2023. <https://doi.org/10.1145/3576899>

A cikkben foglaltak alapján a lineáris blokkláncok lassú tranzakció validációval, alacsony áteresztőképességgel és gyenge skálázhatósággal rendelkeznek, melyek gátolják a szélesebb körű elterjedésüket. A DAG-alapú rendszerek viszont sok esetben jobb teljesítménnyel rendelkeznek, mivel kevesebb kommunikációs, számítási és adattárolási költséget igényelnek. Azonban a fejlettségüket tekintve még nem eléggé kiforrottak, a szükséges fejlesztési irányaik sem teljesen tisztázottak. Persze a nagyobb skálázhatóságnak és gyorsabb validációnak/konfirmációnak ára van: az elosztott főkönyv konzisztenciája gyengül, azaz az egyes csomópontokban lévő eltérő verziók száma nagy lehet, ami a rendszer biztonságát veszélyezteti, és újabb támadási felületeket jelenthet.

Az egy érdekes kérdés, hogy a DAG-alapú blokklánc megoldások, mint pl. az IoT eszközök alkalmazásainak támogatására használt IOTA, teljesítményben és biztonságban hogyan viszonyulna a DONS protokollal vezérelt lineáris blokkláncokhoz. Ilyen irányú vizsgálatokat, kísérleteket még nem végeztünk.

"7. Az 5G és azon túli telekommunikációs rendszerek egyik célkitűzése, hogy nagyon nagy számú IoT eszközt kezelhessenek, beleértve azok adatainak továbbítását, esetleges tárolását és feldolgozását. Mivel a jövőbeli rendszerek lényegesen nagyobb adat továbbítási kapacitást és sokkal alacsonyabb késleltetést biztosítanak, ezért alapfunkcióként olyan szolgáltatások is elérhetőek lesznek, mint a V2X (vehicle to everything). Az értekezésben bemutatott IoT eszközöket tartalmazó rendszerek és blokklánc megoldások alkalmazhatók-e távközlési hálózatokban? Milyen további rétegek szükségesek az integrációhoz?"

Válasz: Az IoT-Köd-Felhő rendszerekben az IoT eszközökben (tipikusan a bennük lévő szenzorokban) keletkezett adatokat juttatjuk el valamilyen kommunikációs protokollon keresztül a ködben és/vagy felhőben futó szolgáltatásokhoz, melyek tárolják és feldolgozzák azokat. Az így végbemenő kommunikáció, azaz üzenetküldés, történhet vezetékes vagy vezeték nélküli hálózatokon keresztül is, tehát a távközlési hálózatokat alkalmazzák a valós IoT-Köd-Felhő rendszerek. Az általunk fejlesztett szimulációs megoldásokban (pl. DISSECT-CF-Fog) absztraháljuk ezeket a kommunikációs protokollokat, legtöbbször csak az egyes kapcsolatok áteresztőképességét, illetve késleltetését tarjuk nyilván. A telekommunikációs rendszerek fejlődésével (pl. 5G és 6G elterjedése) a nagyobb adattovábbítási kapacitás teljesítmény javulást, azaz gyorsabb kommunikációt fog eredményezni, ami valóban elősegítheti újabb alkalmazások megjelenését, mint pl. a V2X rendszerek. A jelentősen nagyobb számú IoT eszközök megjelenése viszont újabb azonosítási és megbízhatósági kérdéseket vet fel, melyeket blokklánc megoldások integrációjával jól kezelhetünk, melyre a következő tanulmány is rámutat:

- M. Kim, I. Oh, K. Yim, M. Sahlabadi and Z. Shukur. Security of 6G-Enabled Vehicle-to-Everything Communication in Emerging Federated Learning and Blockchain Technologies. IEEE Access, vol. 12, pp. 33972-34001, 2024. <https://doi.org/10.1109/ACCESS.2023.3348409>

A fentiek alapján az IoT-Köd-Felhő rendszerek eleve alkalmaznak távközlési hálózatokat az adattovábbítás megoldására, az újabb protokollok megjelenésével az azokra történő átálláshoz az IoT eszközök és az adatokat fogadó gateway szolgáltatások szoftverfrissítésére lehet szükség, illetve akár hardveres eszközök beillesztésére is. Az általunk fejlesztett szimulátorok esetében csupán az újabb protokollok paramétereit alapján szükséges módosítanunk a vizsgálandó használati eseteket. A blokklánc szolgáltatások IoT-Köd-Felhő rendszerekbe történő integrációjához természetesen az azokat működtető infrastruktúrát is ki kell építeni (melyre tekinthetünk újabb rétegeként), mely hatással lehet a teljes rendszer viselkedésére, teljesítményére. Eldöntendő tervezési kérdés ekkor, hogy ez a blokklánc réteg a felhő, köd vagy a végfelhasználói infrastruktúrán kapjon helyet. Ennek megválaszolására dolgoztuk ki a FoBSim szimulátort, mellyel egy konkrét alkalmazás igényeinek megfelelően vizsgálhatjuk meg egy adott méretű és tulajdonságú blokklánc rendszer teljesítményét. Erről a témáról az értekezés 4.2.1-es fejezetében olvashatunk (valamint a [26]-os hivatkozással jelölt cikkben).

"8. A doktori munka számos rövidítést tartalmaz. Az olvasást nagymértékben megkönnyítené egy jelölés összefoglaló."

Válasz: Elfogadom az Opponens kritikáját, valóban sok rövidítés szerepel az értekezésben. Szintén a terjedelmi korlátokat tartottam szem előtt az összesítő részek tervezésében. Egyetértek, hogy az ábrák és a táblázatok listája után a rövidítések listája is helyet kaphatott volna a doktori műben a navigáció elősegítésére.

Szeged, 2024. május 10.


Kertész Attila