Mobil Szimulációs Környezet Felhő-alapú IoT Rendszerek Vizsgálatára

Doktori (PhD) értekezés tézisei

Pflanzner Tamás

Témavezető:

Dr. Kertész Attila

Informatika Doktori Iskola
Szegedi Tudományegyetem

Szeged
2020

Bevezetés

A számítási felhők [8] széles körben elterjedtek és megbízható megoldást nyújtanak az előző évtized problémáira skálázható, virtualizált adattárolási és számítási szolgáltatások segítségével. A publikus felhő szolgáltatók és a különböző középréteg implementációk közötti együttműködés elősegítésére a felhő föderációkat dolgoztak ki és hoztak létre [7]. Az információ- és kommunikációtechnológia fejlődésének egyik legújabb területe a Dologok Internete (Internet of Things, IoT), melyet hatalmas mennyiségű okoseszköz interneten való megjelenése jellemez. Gartner [13] előrejelzések szerint 2020-ban 30 milliárd eszköz állandóan, 200 milliárd eszköz pedig időszakosan fog az internetre csatlakozni. A Huawei becslése szerint 100 milliárd új eszköz felkapcsolódására lehet számítani 2025-ig. Erre az új trendre reagálva, sok felhő szolgáltató kezdett el támogatást nyújtani az IoT rendszerek kezelésére. Az eddigi tapasztalatok alapján bizonyosságot nyert, hogy a felhő szoláltatások képesek kielégíteni az IoT rendszerek elvárásait az adatok feldolgozásával és vizualizációjával kapcsolatban. Gubbi és társai, az IoT rendszereket az elsők között leíró ausztrál kutatócsoport már 2013-ban is hangsúlyozta az IoT rendszerek és a felhők kapcsolatát [10]. Botta és társai pedig bemutatták a két terület együttműködésének főbb tulajdonságait, jellemzőit és a felmerülő problémákat [5], valamint további kutatások ismertették az IoT-felhő rendszerek gyakorlatban történő megvalósítása közben felmerülő számtalan kihívást [12]. Ezek a kutatások is motivációként szolgáltak kutatásaimhoz, mivel felvetették a számtalan protokol és adatformátum kezelésének nehézségeit.

Az IoT rendszerekkel foglalkozó szakembereknek speciális eszközöket kell beszerezniük, amelyeket csatlakoztatniuk kell a hálózatra és be kell üzemelniük őket. Emellett ki kell választaniuk egy megfelelő infrastruktúra szolgáltatót, mely támogatja a szükséges protokollokat és adatformátumokat. Ezekben a kihívásokban próbál segítséget nyújtani az általam kidolgozott hibrid szimulációs rendszer. A rendszer három fő komponensből épül fel, melyek munkám főbb hozzájárulását képviselik: (i) egy MobIoTSim nevű Android-alapú mobil IoT eszközszimulátor, mely valós eszközök viselkedését képes modellezni, (ii) egy testreszabható felhős gateway alkalmazás az IoT eszközök kezeléséhez, az eszközöktől érkező adatok fogadásához, feldolgozásához és megjelenítéséhez; és (iii) egy SUMMON nevű IoT adattároló szolgáltatás, mely valós IoT környezetek adatait gyűjti be és teszi elérhetővé.

A kutatásaim egyik fő motivációját az szolgáltatta, hogy több felhő szolgáltató kezdett el IoT-specifikus szolgáltatásokat kínálni, hogy megkönnyítse az IoT-felhő rendszerek fejlesztését, azonban a megvalósítás nehéz olyan esetekben, ahol egyszerre több dolgot kell működtetni és kezelni. Ilyen esetekre lehet példa a különböző okosváros alkalmazások fejlesztése SIGFOX [3] vagy a LoRa [2] technológiák használatával, melyek esetében különösen költséges és időigényes a valós eszközök kitelepítése, hiszen egy bázisállomás beüzemelése több ezer eurós költséggel és hatalmas mennyiségű beállítási teendővel jár. Ezekre a kihívásokra nyújtok megoldást az értekezésemben IoT eszközök szimulációjára alkalmas szolgáltatások tervezésével és fejlesztésével, valamint a szimulált eszközöket valós felhő szolgáltatásokkal való kezelésével.

Ami a jelenlegi megoldásokat illeti, az elosztott rendszerek vizsgálatára szolgáló meglévő szimulátorok egy része általános hálózati szimulátor, míg egy másik része IoT-specifikus szimulátor, melyek az említett problémákkal is foglalkoznak. Chernyshev és társai átfogó felmérést készítettek az IoT szimulátorokról 2017-ben [9]. Kiemelték, hogy különösen nehéz ebben a témában a kutatás és a meglévő szimulátorok általában csak az architektúra egyetlen rétegére koncentrálnak. Miután megvizsgáltam ezeket az IoT szimulátorokat, én is készítettem egy bővebb felmérést, melyben kiemeltem az egyes szimulátorok különbségeit. Az eredményeket az 1. táblázatban foglaltam össze. Arra a következtetésre jutottam, hogy egy olyan szimulátorra lenne szükség, ami mobilis, így különböző helyekről képes szimulációkat végezni. A fő célja egy ilyen mobil eszközszi-

mulátornak, hogy segítse az olyan felhő alkalmazás fejlesztőket, akik meg szeretnének ismerkedni az IoT rendszerek kezelésével, valós eszközök megvásárlása nélkül. Emellett a tesztelésben és komplex IoT alkalmazások kipróbálásában is segítséget nyújt.

1. táblázat. Szimulátorok összehasonlítása

Szimulátor	**Felhő szolgáltatás**	**IoT szenzorok /eszközök**	**Mesterségesen gen. szenzor adat**	**Valós szenzor /eszköz adat**	**Betanulási idő**	**Grafikus felület**
NetSim	virtuális	nem	igen	nem	hosszú	igen
Qualnet	virtuális	nem	igen	nem	hosszú	igen
OMNet++	virtuális	nem	igen	nem	hosszú	korlátozott
SimIoT	virtuális	igen	igen	nem	hosszú	korlátozott
IoTSim	virtuális	igen	igen	nem	hosszú	nem
iFogSim	virtuális	igen	igen	nem	hosszú	igen
DISSECT-CF	virtuális	igen	igen	igen	hosszú	nem
SimpleIoTSim.	valós	igen	igen	korlátozott	rövid	igen
Atomiton	n.a.	igen	igen	n.a.	rövid	igen
MobIoTSim	valós	igen	igen	igen	rövid	igen

1. ábra. A javasolt keretrendszer IoT-felhő rendszerek szimulációjához

Új tudományos eredmények

Az általam javasolt IoT-felhő szimulátor architektúrája az 1. ábrán látható, mely a hagyományos szimulátorokkal ellentétben egy hibrid szimulációs környezetet definiál. A valósághűbb szimulációk létrehozása céljából elkezdtem valódi szenzoradatokat gyűjteni nyilvános IoT rendszerekből, és ezeket elérhetővé tettem egy adatároló szolgáltatás (I.) segítségével, melyet a MobIoTSim-mel

(II.) összekötve valós eszközök viselkedését lehet szimulálni. Továbbá gateway szolgáltatásokat (III.) is fejlesztettem az IoT eszközök kezeléséhez, az adatok feldolgozására és megjelenítésére. Ezek a gatewayek nyilvános vagy privát felhő vagy köd szolgáltatóknál is telepíthetők.

I. Tézis. Megterveztem és megvalósítottam a MobIoTSim nevű mobil IoT eszközszimulátort, amely felhaználóbarát módon képes IoT eszközöket szimulálni, és a generált adatokat a valós hálózaton továbbítani. Elemeztem a használhatóságát, és kiértékeltem a skálázhatóságát több eszköz egyidejű szimulációjával.

Az általam javasolt mobil IoT eszközszimulátor egy Android operációs rendszerre készített mobilalkalmazás, amely képes több száz IoT eszköz egyidejű szimulálására. A szenzoradatok generálására lehetőség van mesterségesen, a felhasználó által megadott intervallumból véletlenszerű értékek generálásával. Továbbá lehetőség van valós IoT adatokat betölteni és lejátszani a szimuláció során a valósághűbb modellezés érdekében. Az eszközök adatküldési gyakorisága mindkét esetben tetszőlegesen állítható. Az alkalmazás MQTT protokoll használatával küld üzeneteket az Eclipse Paho nyíltforrású programkönyvtár segítségével. Az MQTT protokoll egy IoT-specifikus protokoll [11], mely széles körben támogatott az IoT-felhő alkalmazások körében. Az üzenet tartalma JSON formátumban van kódolva, az IBM IoT Foundation üzenetküldési formátumával kompatibilis módon.

2. ábra. A MobIoTSim Android alkalmazás felhasználói felülete

A szimulátor általános használata során először egy felhőhöz kell csatlakoztatni azt, ahová az adatokat szeretnénk küldeni. Ezt követően lesz lehetőségünk létrehozni és beállítani a szimulálni kívánt eszközöket, majd elindítani az eszközök szimulálását. Ezeket a lépéseket az alkalmazás három fő része valósítja meg: a *Felhő beállítások* (Cloud settings), az *Eszközök* (Devices) és az *Eszköz beállítások* (Device settings) nézetek.

Az *Eszközök* (Devices) nézetben (a 2. ábra bal oldalán) új eszközöket lehet létrehozni vagy importálni, illetve a meglévő eszközök listáját lehet megtekinteni, ahol minden sor egy eszköz vagy eszköz csoportot jelöl. Ezek az elemek a felhasználó által indíthatók el és állíthatók le, külön-külön vagy együttesen is. Az új eszköz hozzáadása és a meglévő eszközök szerkesztése az *Eszköz beállítások* (Device settings) (vagy *Eszköz szerkesztése* Edit device) nézetben lehetséges (a 2. ábra jobb oldalán). Ha a felhasználó egy eszköz sablont használ, akkor a *Típus* (Type) mező, az üzenet tartalma és a frekvencia egy előre definiált értéket vesz fel. Például a *Termosztát* (Thermostat) sablon egy hőmérséklet paraméterrel rendelkezik, és az eszköz bekapcsol, ha elér egy előre meghatározott hőmérsékletet, valamint kikapcsol, ha egy másik, magasabb hőmérsékletet érzékel.

A szimulált eszközökhöz való szenzoradatok generálása többféleképpen is megtörténhet. Alapértelmezetten egy egyedi eszköz létrehozásakor a felhasználó által magadott intervallumból véletlenszerűen kerül egy érték kiválasztásra (alapértelmezetten a *Véletlenszerű adatgenerálás* (Generate random data) opció van kiválasztva). Az egyszerűség kedvéért a felhasználó használhat eszközsablonokat is, melyeknél előre beállított paraméterek és intervallumok szerepelnek, az eszköz típusának megfelelően. A *Véletlenszerű adatgenerálás* (Generate random data) opció kikapcsolásakor a felhasználó kiválaszthat egy fájlt, melyből szenzoradatok tölthetők be. Ezzel a módszerrel korábbi szimulációk is újra elvégezhetők. Továbbá ezzel a módszerrel akár valós IoT alkalmazásokból származó adatokat is be lehet tölteni. Az OpenWeatherMap nevű publikus időjárási adatokat nyújtó szolgáltatás példafájlja is használható, mely több város különféle paramétereit tartalmazza, például a hőmérsékletet, páratartalmat, légnyomást és szélsebességet. A fájlbetöltő funkció segítségével a MobIoTSim képes valós eszközök valósághű modellezésére is.

3. ábra. A küldött üzenetek és a kihasznált szálak közti kapcsolat

Megvizsgáltam a MobIoTSim alkalmazást a szimulált eszközök skálázhatósága szempontjából. Erre a célra készítettem egy egyszerűsített verziót a MobIoTSim-ből (MQTTDemo néven), mely csak az eszközkezeléshez szükséges funkciókat tartalmazza, így automatikusan, szkriptek

használatával történhet az üzenetgenerálás. Olyan alacsonyszintű beállítások is elérhetővé váltak igy módon, mint a szimulált eszközök által használt szálak száma. Továbbá ez a megoldás részletes statisztikai információt is gyűjt a szimulációról, és méri az egyes funkciók végrehajtásához szükséges időt. Képes az IBM Cloud felhőbe telepített gatewayre csatlakozni és az MQTT protokoll segítségével üzeneteket küldeni neki. Amikor a kísérleteket végrehajtottam, a szimulált eszközök száma 20-ra volt korlátozva az IBM Cloud platform által. A beállításokban a szimulált eszközök számát, az adatküldés gyakoriságát, a használt szálak számát és az üzenet típusát és tartalmát lehet beállítani. Az üzenet típusa lehet véletlenszerű paramétereket tartalmazó egyszerű JSON objektum, vagy használható az OpenWeatherMap több városról időjárási adatokat tartalmazó archív fájlja, mint forrás. A többi beállítás beégetésre került a kiértékeléshez, ahogyan az MQTT broker elérhetősége is.

A vizsgálati eredmények a 3. ábrán azt mutatják, hogy a véletlenszerű adatgenerálás elenyésző időt igényel, tehát nem érinti hátrányosan a szimulációkat. Az IBM Cloud által engedélyezett maximális 20 eszköz szimulálása esetén 8 szál használatával volt optimális a teljesítmény.

II. Tézis. Megterveztem és megvalósítottam egy általános célú felhő gateway szolgáltatást, amely bármely felhőbe telepíthető, és IoT eszközök kezelésére képes a tőlük származó adatok fogadásával, tárolásával és feldolgozásával vagy megjelenítésével. Validáltam a MobIoTSim-el való együttes használhatóságát, és kiértékeltem a skálázhatóságát.

4. ábra. A MobIoTSim eszközcsoportoktól származó részletes, paraméterenkénti adatvizualizáció az IBM Cloud felhőben lévő privát gateway szolgáltatásban

2. táblázat. Mérések az időjárás sablonnal.

Eszközök száma	**10**	**100**	**250**	**450**
CPU haszn. (%)	1.59	12.27	29.53	52.29
Memória (MB)	110.07	110.22	110.35	111.05
Hálózat (B/s)	853.6	855.16	881.34	890.66
Üzenet méret (KB)	2468	24695	61666	111110
Üzenet db	6000	60046	149940	270165

Miután megvizsgáltam több különböző felhő szolgáltatás IoT megoldásait, feltérképeztem a követelményeit egy könnyen használható és szimulált, valamint valós IoT eszközök kezelésére is alkalmas gatewaynek. Ezek alapján kifejlesztettem egy testreszabható gatewayt, amely szinte bármelyik felhő szolgáltatónál telepíthető, hogy akár több száz különféle eszközt lehessen egyszerre kezelni.

Ez az általános gateway egy node.js alkalmazás, amely képes egyszerre több eszköz kezelésére MQTT protokoll segítségével, és egy grafikus felületen jeleníti meg a MobIoTSim-ből érkező szenzoradatokat, ahogy a 4. ábra mutatja. A szimulált eszközöktől érkező üzeneteket (JSON formátumban) egy MQTT szerver fogadja. Lehetőség van a szimulált eszközök felé is üzenetet (értesítést) küldeni. A valós idejű diagram rajzolásánál az eszközök csoportosításával egyetlen diagramon is áttekinthető az összes eszközről küldött adat. Habár az IBM Cloud felhő is mutat némi erőforrás-használati információt, Docker konténerek használatával részletesebb adatok érhetőek el az alkalmazással kapcsolatban. Ecélból egy Docker konténerbe csomagoltam a továbbfejlesztett gatewayt, így növeltem hordozhatóságát.

5. ábra. Levélméret változásának vizualizációja az IoLT Okos Cserép Gatewayben

Megvizsgáltam ennek az általános gatewaynek a skálázhatóságát. Kitelepítettem a Docker konténerbe csomagolt gatewayt az IBM Cloud felhőbe, majd több kísérletet végeztem el. Az első körben a MobIoTSim termosztát sablont használtam, hogy 900 szimulált eszközt vizsgáljak. Az

eredmények azt mutatták, hogy az alkalmazást csak kis mértékben terhelte le a nagy számú eszköz kezelése. A második körben a processzorra külön terhelést helyeztem (egy nagyobb számítási igényű algoritmussal) és megismételtem a méréseket 10, 100, 250 és 400 eszközzel. Az eredmények azt mutatták, hogy a dupla mennyiségű üzenettel (azaz 0.5 másodpercenkénti üzenetküldéssel), a processzor használata is duplájára emelkedett. Az utolsó körben a MobIoTSim időjárás sablonját használtam. Mivel a MobIoTSim képes fájlból beolvasott adatok alapján eszközöket szimulálni, így ezt a funkciót használtam, hogy az OpenWeatherMap adatokat használja. Mind a négy eszközcsoportra újra lefuttattam a teszteket egy másodperces frekvencián. A részletes eredmények a 2. táblázatban láthatók. Az eredmények azt mutatták, hogy az időjárási adatokat tartalmazó nagyobb méretű üzenetek nagyobb processzor terheltséget idéztek elő.

Az okos agrárium is egy gyorsan növekvő terület az okos rendszereken belül. A jelenkori információ- és kommunikációtechnológia fejlődésnek köszönhetően az IoT technológiák segíthetik ezt a területet is, melyet megfizethető fenotipizálásnak (affordable phenotyping) neveznek. Ehhez kapcsolódóan részt vettem az Élő Dolgok Internete (Internet of Living Things) projektben, ahol IoLT Okos Cserép (IoLT Smart Pot) néven megfizethető fenotipizálási platformot készítettünk kis méretű növények számára, amely képes a növények növekedésének távoli megfigyelésére egy átlagos üvegházban. Egy ilyen okos cserepekből álló klaszter kezeléséhez az IoLT Okos Cserép Gateway (IoLT Smart Pot Gateway) alkalmazást terveztem meg, és vettem részt kifejlesztésében az általános gateway fejlesztési eredményeim és tapasztalataim alapján. Ezt követően részt vettem a megoldás szimulációs skálázhatósági vizsgálatainak elvégzésében, és megmutattam helyes működését valós környezetben. Az 5. ábrán a specializált gateway webes felületének képernyőképe látható, ahol a generált diagram a valós szenzoradatok alapján egy, a klaszterben lévő növény leveleinek növekedését mutatja. A görbén jól megfigyelhetők a levélmozgás cirkadián ritmusai (a sötétségre és a fényre reagálva).

III. Tézis. Megterveztem és megvalósítottam a SUMMON nevű IoT adattároló szolgáltatást, amely képes valós IoT alkalmazásokból rendszeresen adatot gyűjteni és tárolni, akár az ezt közvetlenül nem támogató alkalmazásoktól is. Megmutattam hatékony működését három használati eseten keresztül. Az összegyűjtött adathalmazokat elérhetővé tettem a SUMMON szolgáltatás keretében, melyek valósághűbb IoT szimulációkat tesznek lehetővé.

Az Európai Bizottság 2016-ban kiadott egy útmutató dokumentumot [1] melyben a Horizon 2020 projektek által generált kutatási adatok nyílt hozzáférhetőségét és digitális újrahasználhatóságát ösztönözte. Ebben a dokumentumban definiálták az úgynevezett FAIR adatkezelési irányelveket, hogy segítsék a Horizon 2020 résztvevőket a kutatási adatok megtalálásában, hozzáférésében és újrahasznosításában. Ezt az útmutatót követve tűztem ki célul, hogy egy nyílt IoT adattároló szolgáltatást készítek, mely az adatok közti keresést, hozzáférést és újrahasználhatóságot segíti elő IoT-felhőkkel kapcsolatos szimulációs vizsgálatokhoz.

A MobIoTSim első verzióiban csak mesterségesen generált értékek voltak használhatóak a szenzor szimulációk során. Később a fájlbetöltés funkcióval bővítettem az alkalmazást, hogy könnyebben konfigurálható vizsgálatokat lehessen végezni és valós IoT adatokat is fel lehessen használni a szimulációkhoz. Létrehoztam egy időjárási sablont, amely okos városok időjárásfi-

gyelő eszközeit reprezentálja, amihez az OpenWeatherMap szolgáltatás adatait használtam. A további alkalmazási területek támogatásához egy nyílt IoT adattároló szolgáltatást terveztem használni a MobIoTSim-mel együtt. Ezért fejlesztettem ki a SUMMON (Smart system Usage data Management and MONitoring) nevű megoldást, ami egy adatok összegyűjtésére, szűrésére és tárolására alkalmas szolgáltatás. A szolgáltatásban kilistázhatóak az elérhető adathalmazok, melyek lekérdezés és szűrés után a MobIoTSim szimulátorba tölthetőek nagyméretű vizsgálatokhoz. Több okos város valós adatait gyűjtöttem össze, így például a SmartME [6] és a CityPulse [4] projektekből.

6. ábra. IoT-felhő rendszerek vizsgálatához kidolgozott kiterjesztett szimulációs környezet

Az okos városok adatainak összegyűjtése és szűrése mellett, olyan közösségi módon működő szolgáltatásoktól is fontos lehet adatokat kinyerni a szimulációs környezetek számára, melyek dinamikusan frissülő weboldalakkal rendelkeznek és közvetlenül nem osztják meg a begyűjtött és megjelenített adatokat. Az ilyen weboldalakból történő adatok kinyeréséhez webes feltérképezés (web crawling) módszerét javasoltam az IoT adatok kigyűjtéséhez és újrahasználásához. Ez a javaslat kiegészíti a korábbi megoldást, és a webes feltérképező szolgáltatás segítségével további adathalmazokkal bővíti a SUMMON adattár szolgáltatást.

3. táblázat. A három használati eset összehasonlítása

Projekt	Idő (sec)	Adat (KB)	Üzenetek száma
idokep	4	20.24	50
livetraffic	30	17.7	47
parkolás	10	7.48	33

Részt vettem a javasolt webes feltérképező szolgáltatás kifejlesztésében, mely során a Scarpy nevű nyílt forráskódú python-alapú keretrendszert használtuk az adatok kinyerésére és a Scrapyd nevű alkalmazást a webes feltérképező pókok (spider) párhuzamos kezelésére. A szolgáltatás prototípusát egy Docker konténerbe csomagoltuk, majd a Google felhőbe telepítettük és a SUMMON prototípusával kötöttük össze egy REST API segítségével, hogy megvizsgáljuk az együttműködésüket. Ahhoz, hogy megkapjuk a kívánt értékeket különböző keresési és szűrési módokat kellett

alkalmaznunk a weboldal HTML forrásán. Mielőtt a MongoDB adatbázisba mentenénk az adatokat, a felhasználóneveket álnevesítettük adatvédelmi okokból. Így a webes feltérképező szolgáltatás által kinyert adatok rendszeresen lekérhetők és tárolhatók a SUMMON-ban. A megoldás kifejlesztése után részt vettem három okos város alkalmazás feltérképezésének használhatósági vizsgálatában: a felhasználói jelentések alapján működő Idokep.hu szolgáltatásának, a valós idejű közlekedési híreket szolgáltató Automobile Association Developments Ltd. alkalmazásának, és a felhő alapú ParkMe parkoló kezelő szolgáltatásának elemzésében. A 3. táblázat összegzi a három használati eset egy napon történt kiértékelése közben végzett adatgyűjtések során mért átlagos futási időket, a mentett adatok nagyságát és számát. Az eredmények azt mutatták, hogy az egyes okos rendszerek weboldalainak elrendezése nagyon eltérő lehet, így egy általánosított pók (azaz feltérképező algoritmus) nem tudná egységesen elemezni őket (tehát egyedi megoldásokra van szükség).

Összefoglalás

Napjainkban a hatalmas számú új eszköz interneten való megjelenése az IoT paradigma megszületését eredményezte. Az újonnan felmerülő igényekre a felhő szolgáltatók is felfigyeltek és új, IoT-specifikus szolgáltatásokat kezdtek kínálni. Az IoT rendszerekkel foglalkozó szakemberek több kihívással is küzdenek: okoseszközöket kell vásárolniuk, meg kell tervezzék és meg kell valósítsák az azokat kezelő infrastruktúrát, IoT alkalmazásokat kell fejleszteniük, tesztelniük kell ezeket az alkalmazásokat, ki kell értékelniük, végül finomhangolni kell azokat.

Ebben az értekezésben olyan eszközöket dolgoztam ki, melyek az említett kihívások közül néhányat segítenek leküzdeni. Kifejlesztettem egy komplex, hibrid szimulációs környezetet, amely a következő komponensekből áll: egy MobIoTSim nevű Android-alapú mobil eszközszimulátor, amely képes több száz IoT eszköz egyidejű szimulációjára; egy testreszabható felhő gateway szolgáltatás, amely képes kezelni a szimulált IoT eszközöket; és egy SUMMON nevű IoT adattároló szolgáltatás, amely valós IoT alkalmazásoktól gyűjt adatokat, hogy ezzel valósághűbb szimulációt segítsen elő. Bemutattam a használatát ennek a komplex szimulációs környezetnek, és kiértékeltem a skálázhatóságát. Meggyőződésem, hogy ezek az eszközök hozzájárulnak az IoT rendszerek költséghatékonyabb tervezéséhez, fejlesztéséhez és kiértékeléséhez, mind akadémiai kutatócsoportok, mind ipari szereplők számára.

A tézisek tudományos eredményeit számos nemzetközi folyóiratban, konferencia és workshop cikkben publikáltam, és különféle tudományos fórumokon adtam elő. A disszertáció publikációi több későbbi kutatás alapjául szolgáltak, tudományos együttműködéseket hoztak létre, amelyet a sok független hivatkozás is mutat, melyeket a 4. táblázatban összesítettem.

Az értekezésben bemutatott kutatási eredmények nagy része az Internet of Living Things (GINOP-2.3.2-15-2016-00037), a Smart Systems Research Institute (20391-3/2018/FEKUSTRAT és TUDFO/47138-1/2019-ITM), az EFOP-3.6.1-16-2016-00008, és az OTKA FK 131793 hazai, és az EU COST Action IC1304 (ACROSS) és EU COST Action CA17136 (INDAIRPOLLNET) nemzetközi projektekben való aktív részvétel során keletkeztek.

A bemutatott hibrid szimulációs környezet komponensei folyamatos fejlesztés alatt állnak, és a forráskódjuk bárki számára elérhető a GitHub-on az alábbi linkeken:

- https://github.com/sed-szeged/MobIoTSim/
- https://github.com/sed-szeged/MobIoTSimBluemixGateway
- https://github.com/sed-szeged/iotgateway

- https://github.com/sed-inf-u-szeged/IoLT-Smart-Pot-Gateway
- https://github.com/sed-szeged/SUMMON

4. táblázat. Publikációk, Tézisek és független hivatkozásaik

	Thesis I.	Thesis II.	Thesis III.	Független hivatkozások	
				Google Scholar	MTMT
[P14]		♦		9	4
[P12]		♦			
[P13]		♦		1	
[P6]	♦	♦		35	21
[P1]			♦	1	
[P10]	♦	♦		12	6
[P9]	♦	♦	♦	1	1
[P4]	♦	♦		2	
[P7]	♦		♦	5	
[P11]	♦	♦	♦	1	1
[P8]	♦		♦	1	1
[P2]	♦	♦	♦	12	8
[P3]			♦		
[P5]		♦			
Sum	8	10	7	80	42

Köszönetnyilvánítás

Az elmúlt öt évben lehetőségem volt egy olyan kutatócsoportnak a tagja lenni, amely a felhő és IoT technológiák együttműködésében kezdett el szaktudást felhalmozni és ezen összetett rendszerek vizsgálatához szimulációs eszközöket biztosítani. Ennek következménye, hogy az értekezésben bemutatott eredményeket nem egyedüli kutatóként, hanem a kutatócsoport tagjaként értem el. Hálás vagyok a kollégáimnak, hogy biztosították ezeket az együttműködési lehetőségeket.

Külön szeretném megköszönni témavezetőmnek, Dr. Kertész Attilának az iránymutatást, a hasznos tanácsokat, az időt és energiát, amit rám fordított. Hálával tartozom, amiért elindított a kutatói pályán és, hogy mindig nagyszerű élményekkel gazdagodtam a közös szakmai utakon.

Meg szeretném továbbá köszönni Prof. Dr. Gyimóthy Tibornak, a Szoftverfejleszés Tanszék egykori tanszékvezetőjének, valamint Dr. Ferenc Rudolfnak, a Szoftverfejleszés Tanszék jelenlegi tanszékvezetőjének, hogy segítették a PhD tanulmányaim előmenetelét. Társszerzőimnek is köszönettel tartozom, hogy rengeteg jó ötlettel szolgáltak és segítettek a technikai akadályok leküzdésében, valamint Szűcs Editnek is köszönom a disszertációm angol nyelvű változatának lektorálását.

Köszönettel tartozom kollégáimnak, különösképpen Márkus Andrásnak, Dr. Tóth Zoltánnak, Dr. Bán Dénesnek és Dr. Kádár Istvánnak, hogy számíthattam rájuk.

Végül, de nem utolsó sorban szeretném megköszönni szüleimnek a gondoskodást és bíztatást. Nagyon szerencsésnek érzem magam, hogy családtagjaim jó példát mutatnak nekem: anyukám megbízhatóságával, apukám fáradhatatlanságával és széles érdeklődési körével, nagymamáim rátermettségükkel és kitartásukkal, tatáim elhivatottságukkal és életvidámságukkal, nagybátyám

kreatív gondolkodásával, keresztanyám jószívűségével, keresztapám talpraesettségével. Szívből köszönöm Katának, hogy mellettem van a küzdelmes pillanatokban, hogy társam a csodálatos élményekben, de legfőképpen, hogy velem van az egyszerű mindennapokban is. Hálás vagyok Dórinak és Daninak, hogy testvéreim helyett testvéreim. Köszönöm barátaimnak a vidám pillanatokat, a szeretetet és megértést.

Az ebben az értekezésben bemutatott kutatások a GINOP-2.3.2-15-2016-00037 azonosítójú, Internet of Living Things című projekt keretében készültek, melyet a Magyar Kormány és az Európai Regionális Fejlesztési Alap támogatott.

Pflanzner Tamás, 2020

Hivatkozások

[1] European Commission, Guidelines on FAIR Data Management in Horizon 2020. Version 3.0, July 2016. Online: http://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/-h2020-hi-oa-data-mgt_en.pdf.

[2] LoRa Technology. `https://www.lora-alliance.org/What-Is-LoRa/Technology`, 2017. Accessed: 2017-03-20.

[3] SIGFOX. `http://www.sigfox.com/en/#!/connected-world`, 2017. Accessed: 2017-03-21.

[4] Stefan Bischof, Athanasios Karapantelakis, Cosmin-Septimiu Nechifor, Amit P Sheth, Alessandra Mileo, and Payam Barnaghi. Semantic modelling of smart city data. 2014.

[5] Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. Integration of cloud computing and internet of things: a survey. *Future generation computer systems*, 56:684–700, 2016.

[6] Dario Bruneo, Salvatore Distefano, Francesco Longo, Giovanni Merlino, and Antonio Puliafito. Turning messina into a smart city: The# smartme experience. In *Smart Cities in the Mediterranean*, pages 135–152. Springer, 2017.

[7] Wojciech Burakowski, Andrzej Beben, Hans van den Berg, Joost W Bosman, Gerhard Hasslinger, Attila Kertesz, Steven Latre, Rob van der Mei, Tamas Pflanzner, Patrick Gwydion Poullie, et al. Traffic management for cloud federation. In *Autonomous Control for a Reliable Internet of Services*, pages 269–312. Springer, Cham, 2018.

[8] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.

[9] Maxim Chernyshev, Zubair Baig, Oladayo Bello, and Sherali Zeadally. Internet of things (iot): Research, simulators, and testbeds. *IEEE Internet of Things Journal*, 5(3):1637–1647, 2017.

[10] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[11] N. Naik. Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In *2017 IEEE International Systems Engineering Symposium (ISSE)*, pages 1–7, 2017.

[12] Stefan Nastic, Sanjin Sehic, Duc-Hung Le, Hong-Linh Truong, and Schahram Dustdar. Provisioning software-defined iot cloud systems. In *2014 international conference on future internet of things and cloud*, pages 288–295. IEEE, 2014.

[13] B Rossi. Gartner's internet of things predictions. *Information Age, Vitesse Media, January*, 2015.

Publikációk

[P1] Attila Kertesz and Tamas Pflanzner. Towards enabling scientific workflows for the future internet of things. In *Internet of Things. IoT Infrastructures. Second International Summit, IoT 360° 2015, Rome, Italy, October 27-29, 2015. Revised Selected Papers, Part I*, 1 2017.

[P2] Attila Kertesz, Tamas Pflanzner, and Tibor Gyimothy. A mobile iot device simulator for iot-fog-cloud systems. *Journal of Grid Computing*, 17(3):529–551, 2019.

[P3] Tamas Pflanzner, Zoltan Feher, and Attila Kertesz. A crawling approach to facilitate open iot data archiving and reuse. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 235–242. IEEE, 2019.

[P4] Tamas Pflanzner, Marta Fidrich, and Attila Kertesz. Simulating sensor devices for experimenting with iot cloud systems. In *Connected Environments for the Internet of Things*, pages 105–126. Springer, 2017.

[P5] Tamas Pflanzner, Miklos Hovari, Imre Vass, and Attila Kertesz. Designing an iot-cloud gateway for the internet of living things. In *International Conference on Cloud Computing and Services Science*, pages 23–41. Springer, 2019.

[P6] Tamas Pflanzner and Attila Kertesz. A survey of iot cloud providers. In *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 730–735. IEEE, 2016.

[P7] Tamas Pflanzner and Attila Kertész. A taxonomy and survey of iot cloud applications. *EAI Endorsed Transactions on Internet of Things*, 3(12), 10 2017.

[P8] Tamas Pflanzner and Attila Kertesz. Analyzing iot, fog and cloud environments using real sensor data. In *Fog Computing*, pages 83–105. Springer, Cham, 2018.

[P9] Tamas Pflanzner and Attila Kertész. A private gateway for investigating iot data management. In Víctor Méndez Muñoz, Donald Ferguson, Markus Helfert, and Claus Pahl, editors, *Proceedings of the 8th International Conference on Cloud Computing and Services Science, CLOSER 2018, Funchal, Madeira, Portugal, March 19-21, 2018*, pages 526–532. SciTePress, 2018.

[P10] Tamas Pflanzner, Attila Kertesz, Bart Spinnewyn, and Steven Latre. Mobiotsim: towards a mobile iot device simulator. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 21–27. IEEE, 2016.

[P11] Tamas Pflanzner, K Zs Leszko, and Attila Kertesz. Summon: Gathering smart city data to support iot-fog-cloud simulations. In *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 71–78. IEEE, 2018.

[P12] Tamas Pflanzner, Roland Tornyai, Akos Zoltan Goracz, and Attila Kertesz. Characterizing paas solutions enabling cloud federations. In *Application Development and Design: Concepts, Methodologies, Tools, and Applications*, pages 1095–1120. IGI Global, 2018.

[P13] Tamas Pflanzner, Roland Tornyai, and Attila Kertesz. Towards enabling clouds for iot: Interoperable data management approaches by multi-clouds. In *Connectivity Frameworks for Smart Devices*, pages 187–207. Springer, 2016.

[P14] Roland Tornyai, Tamas Pflanzner, Attila Kertesz, Anita Schmidt, and Balazs Gibizer. Performance analysis of an openstack private cloud. In *Proceedings of the 6th International Conference on Cloud Computing and Services Science - Volume 1 and 2*, CLOSER 2016, page 282–289, Setubal, PRT, 2016. SCITEPRESS - Science and Technology Publications, Lda.