



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

Erőforrás-kezelés felhőrendszerekben

MTA DOKTORI DISSZERTÁCIÓ TÉZISEI

Szerző
Toka László

2023. január 26.

Tartalomjegyzék

1. Bevezetés	2
2. Kutatási célkitűzések és módszerek	3
3. A kutatási területek kapcsolatai és azok háttere	3
4. Új tudományos eredmények	6
4.1. A felhők közötti üzleti kapcsolatok	6
4.2. Felhőn belüli erőforráskezelési és vezénylési módszerek	11
4.3. A felhőhozzáférés erőforrás-kiosztása	16
5. Az eredmények hasznosíthatósága és hatása	18
5.1. Szolgáltatók közötti árképzés	18
5.2. Felhőkezelési technikák	19
5.3. Felhő alatti erőforrás-megosztás	20

1. Bevezetés

A felhőalapú számítástechnika alig két évtized alatt átalakította az információs technológiák színterét. Az elképesztő technológiai fejlődés mind a számítástechnika, mind a hálózatépítés terén számos új alkalmazást és szolgáltatást tett lehetővé, amelyek rendkívül nagy léptékben működnek különböző felhőplatformokon. A nyilvános felhőplatformok, mint például az Amazon Web Services [1], a Google Cloud Platform [2] és a Microsoft Azure [3] igény szerint „tetszőleges” mennyiségű virtuális erőforrást képesek biztosítani virtualizációs technikák és erőforrás-kezelési mechanizmusok felhasználásával. A jól megtervezett adatközpontok minden szükséges fizikai eszközt tartalmaznak, köztük több ezer szervert és hálózati eszközt, és az alkalmazásüzemeltetéssel kapcsolatos terheket is vállalják, biztosítva ezzel a nagy megbízhatóságot és magas teljesítményt. A felhőalkalmazás tulajdonosának nincs más dolga, mint kiválasztani a legmegfelelőbb felhőszolgáltatás-ajánlatot, és az alkalmazását a felhőben üzembe helyezni: ez azt jelenti, hogy i) nem szükséges kezdeti infrastruktúra befektetés, mivel a felhőszolgáltatások használat-alapú fizetési konstrukciókat kínálnak, és ii) nincs szükség a maximális kapacitásra történő tervezésre, mivel az erőforrás-ellátás rugalmas a felhőben, az alkalmazások léptéke általában automatikusan állítható. A felhőbérleti oldalon a nem létező infrastrukturális beruházások mellett az alkalmazások felhőbe történő telepítésének előnyei közé tartozik még az alacsony működési költség. A hatékony működés hátterében az adatközpontok számítási infrastruktúrájának méretgazdaságossága és a számtalan felhőbérlet által megosztott erőforráshalmaz időben multiplexált felhasználása, azaz folyamatos magas kihasználtsága áll.

Az elmúlt évtizedben azonban a központilag elhelyezett adatközpontok helyett a fizikai erőforrások egyre közelebb kerültek a felhasználókhoz, hogy lehetővé tegyék az újszerű, különösen a késleltetésérzékeny alkalmazások működését. Ez a paradigmaváltás megnyitotta a kaput a távközlési szolgáltatók, a mobil- és vezeték nélküli hálózatok szolgáltatói előtt: kissé megkésve ugyan, de igyekeznek csatlakozni a felhő ökoszisztémához, hogy részesei lehessenek a sikertörténetnek. Különböző koncepciók és paradigmák jelentek meg a vásárlók és a végberendezések közelében telepített számítási erőforrások kiaknázásának megfelelő módjára. Peremszámítás, kód számítástechnika, többszozáférésű (korábbi nevén mobil) peremszámítás külön fogalmak, ennek ellenére több közös céljuk és jellemzőjük van. Ezek a koncepciók különböző technológiai és üzleti felhasználási esetekkel foglalkoznak, de a távközlési szereplők minden esetben döntő szerepet játszanak vagy önálló entitásként vagy felhőszolgáltatókkal közösen.

Emellett az újszerű felhőplatformok és az azok által ajánlott képességek gyökeresen átalakították a felhőben futó *szoftvert*, ezzel együtt pedig megváltoztatták az alkalmazandó szoftverfejlesztési technikákat is. A dedikált virtuális gépekben (VM) futó monolitikus alkalmazásokból kiindulva megje-

lentek a mikroszolgáltatások: lazán csatolt, egymással kommunikáló szoftvermodulok, amelyek külön konténerekben, vagy különálló funkciókként futnak, amelyeket a kiszolgáló felhőrendszer kezel. A fejlesztők és az alkalmazásszolgáltatók nyilvánvalóan profitálhatnak ebből a váltásból, azonban felhőplatform oldalán több új kihívás is felmerül. Ezért a kutatói közösség az elmúlt években jelentős erőfeszítéseket tett e témában, és számos elméleti eredmény látott napvilágot a kapcsolódó matematikai problémák különböző aspektusaival és változataival kapcsolatban: számos tudományterület különféle technikáit alkalmazták a matematikai programozástól kezdve a gráfelméleten át a gépi tanulásig.

2. Kutatási célkitűzések és módszerek

A disszertációban bemutatott, felhővel kapcsolatos kutatások célja olyan rendszerek és módszerek tervezése, amelyek a felhőalapú számítástechnika világát még megbízhatóbbá, megfizethetőbbé és könnyebben elérhetővé teszik az ügyfelek számára világszerte. A cél eléréséhez alkalmazott megközelítés kettős: egyrészt a hagyományos távközlési szolgáltatók felhő-üzletágban való részvételét vizsgálom, másrészt felhőkezelési technikákat javaslok a magas szolgáltatásminőség (QoS) érdekében. Mindkét megközelítésben nagy hangsúlyt kap a felhő alapú alkalmazások végpontok közötti késleltetése és az erőforrások gazdaságos felhasználása: előbbi nélkülözhetetlen a jó QoS-hoz, utóbbi pedig még olcsóbbá teszi a felhőalapú alkalmazások üzemeltetését.

A modellezéshez alkalmazott eszközkészlet és elemzési módszertan a matematikai apparátus széles skáláján terül el. A gráfelméletet olyan helyzetek modellezésére használom, ahol bizonyos entitások kölcsönhatása nem csak egymás státuszára van hatással, hanem közvetett hatással van azokra is, amelyek egy ugrásnál távolabb vannak tőlük. A játékelméletet olyan esetekre alkalmazom, amelyekben több racionális entitás lép interakcióba, mindegyik a saját céljait és előre meghatározott stratégiáit követi, emiatt egy elosztott keretrendszer szükségeltetik az egyensúlyi pontok megtalálásához a rendszerben. A gépi tanulási (ML) technikák biztosítják az ügyfelek viselkedésének pontos előrejelzését, ami kiemelten fontos a fent említett céljaink eléréséhez, azaz megbízható, megfizethető és elérhető felhőszolgáltatások biztosításához a kívánt QoS szinten. Továbbá még valószínűségi számítás, sorbánállási modellezést, differenciálegyenlet-rendszereket, dinamikus programozási megfogalmazást és Karp-redukciókat tartalmaznak a bemutatott kutatási fejezetek.

3. A kutatási területek kapcsolatai és azok háttere

A 4.1. fejezetben a több szolgáltató által nyújtott felhőalapú szolgáltatások elképzelhető üzleti struktúráinak jellemzésével kezdjük a vizsgálatot. A

jelenlegi többszintű Internetszolgáltatói (ISP) struktúrát feltételezve a világméretű hálózati szolgáltatásokra kiindulópontnak, megvizsgálom, hogy a jelenlegi ISP struktúra miért és hogyan változhat az új üzleti kapcsolatok mentén, azaz a felhők közötti gazdasági kapcsolatokat vizsgáljuk.

A réteges ISP kapcsolati hálóból kiindulva egy hálózatformáló játékot írok le, amelyben a játékosok az ISP-eket képviselik; egy kapcsolat létrehozása fenntartási költséggel jár ugyanakkor a működési költségek és bevételek a közvetítői szolgáltatásokat igénybe vevő és az azokat nyújtó ISP-k között fizetett közvetítői költségektől is függenek. Munkámat azok kapcsolódó kutatási eredmények ihlették, amelyek az ISP-k közötti növekvő számú egyenrangú kapcsolatok jelenségével foglalkoznak. Az említett munkákkal ellentétben tanulmányom az 5. generációs mobilhálózatok (5G) és a felhő infrastruktúra szolgáltatók közötti megállapodásokra összpontosít, akár közvetlenül, akár közvetítő feleken keresztül jöjjenek azok létre. Ezzel szemben a kapcsolódó kutatásokban a peering vagy tranzit kapcsolatokon keresztül kicserélt internetes forgalmat vizsgálják. Sőt, úgy gondolom, hogy a jelenlegi internetes árazási modellek a szolgáltatások jövőbeni változása miatt megkérdőjeleződnek. Pontosabban, mindaddig, amíg a szolgáltatás pusztán kapcsolat, az internetes forgalom kezelése és a kapcsolati topológia elrejtése a gyakorlat. Ezért ma egy távközlési szolgáltató vagy egy ISP pusztán alapértelmezett hozzáférést biztosít az internet többi részéhez, és csak a komoly szűk keresztmetszetek elkerülése a cél. Az elképzelt 5G-szolgáltatásokban [4] azonban a hálózati kapcsolat összefonódik a számítási szolgáltatásokkal, és a közelség kulcsfontosságú tényezővé válik. Ezért a jövőben az elérhetőségi információkat kiegészíthetik felhőbeli rendelkezésre állási és képességi információk pl. a peremfelhő telepítésekkel kapcsolatban.

Míg az 5G forradalmasította a hálózati szolgáltatásokat [5, 6], a folyamatban lévő és tervezett változások számos más területet is érintenek a felhő/perem számítástechnikától a vertikális iparágakig, beleértve az egészségügyet, az Ipar 4.0-t, és a közlekedést [7]. Az 5G víziókban a gyakran hallott szolgáltatási szintű kulcsszavak a költséghatékonyság és a jobb szolgáltatásnyújtás gyors szolgáltatásteremtéssel, gyors újrakonfigurálással és az ügyfelek széles földrajzi elérésével. Ezt a paradigmaváltást technológiailag a hálózati szolgáltatásvirtualizálás (NFV) [8] (a VM-ekbe telepített távközlési hálózatbeli funkciók már általános célú számítógépeken is futtathatók, ahelyett, hogy drága dedikált hardveren futnának a hagyományos módon) és a szoftvervezérelt hálózatok (SDN) [9] (a hálózati eszközök konfigurálása és vezérlése könnyen kezelhető, gyakran központilag futtatott szoftveralkalmazásokkal [10] történik) teszik lehetővé. Ennek a paradigmaváltásnak a gazdasági vonatkozásait vizsgáljuk: hogyan határozzák meg árakat a szolgáltatók egy ilyen piacon, és az ügyfelek hogyan választják ki a szükséges erőforrásokat ezeken az árakon.

Ezt követően a 4.2. fejezetben meghatározom a felhőerőforrás-kezelés legégetőbb kihívásait, amelyek jellemzően a peremfelhők esetében a lehang-

súlyosabbak, és ezekre kínálok megnyugtató megoldásokat.

A jövőbeli alkalmazások, például a kiterjesztett valóság-alapú alkalmazások vagy az 5G és azon túli távközlési szolgáltatások rendkívül megbízható és alacsony késleltetésű kommunikációt igényelnek a számítástechnikai és hálózati infrastruktúrától. Az 5G gyors hozzáférési hálózatára épülő peremszámítás ilyen szigorú késleltetési feltételeket képes teljesíteni. A távoli peremcsomópontok azonban hajlamosak a meghibásodásokra, és leállási idejük sok esetben hosszabb lehet, mint egy központi infrastruktúráé, pl. az adatközpontoké. Ezért, bár egy szolgáltatás számítási elemeinek a peremfelhőbe történő telepítése minimálisra csökkenti a szolgáltatás késleltetését, a szolgáltatások magas megbízhatóságának biztosítása kihívást jelent.

Az erőforrások dinamikus kiosztása az alkalmazást összetevő VM-eknek vagy konténereknek, és azok igény szerinti méretezése kihívást jelenthet. A méretezési logikát különféle szolgáltatáskezelési célok vezérelhetik, a cél lehet például az erőforrás-használat minimalizálása egy adott QoS cél fenntartása mellett, vagy a QoS vállalások (SLA) megsértésének minimalizálása, függetlenül a kibérelt erőforrásokért fizetett ártól. A léptékkal kapcsolatos döntéshozatalt tovább nehezíti a felhőalkalmazás sajátos skálázási viselkedése, vagyis az a függvény, amely az adott SLA korlátok betartásával a kiszolgáltató felhasználói kérélmek mennyiségét a megbízható működéshez szükséges erőforrások mennyiségére fordítja.

A felhőalapú alkalmazások manapság jellemzően a mikroszolgáltatási architektúrát követik, amennyiben a monolitikus szoftvereket kisebb, egymástól függetlenül menedzselte komponensekre bontják, ezeket utána általában szoftverkonténerekbe telepítik, amelyeket a felhőplatform külön-külön vezényel és skáláz, lehetővé téve az optimális erőforrás-kihasználást [11]. Ez hozzáadott késleltetést okoz az alkalmazás válaszidejében, amelyet kezelni kell a magas QoS fenntartása érdekében.

Ezekkel a felhőn belüli kihívásokkal a 4.2. fejezet megfelelő szakaszai foglalkoznak. Az azt követő 4.3. fejezetben egy további lépést teszek a felhőből lefelé, és a felhő alatti erőforrás-kiosztást tanulmányozom: szolgáltatásminőség-biztosítási keretrendszereket javaslok koordinálatlan és központi optimalizált környezetben. Az első esetben a felhasználók lehetőséget kapnak arra, hogy jelezzék sürgős sávzélesség-igényeiket, és a hálózati hozzáférési pont ezek alapján hozza meg az ütemezési döntéseket. A második esetben egy felhő alapú dolgok internete (IoT) architektúrát mutatok be, amely nagy sávzélességű videó folyamatok között maximalizálja a hasznosságot olyan esetekben, amikor egy szűk feltöltési csatornán a meg kell osztzkodniuk. A javasolt rendszer jelentősen csökkenti a felhasznált feltöltési sávzélességet és a feldolgozáshoz igénybe vett teljesítményt a felhőben.

4. Új tudományos eredmények

4.1. A felhők közötti üzleti kapcsolatok

A jövőben elképzelt 5G ökoszisztéma epicentrumában a számítási és hálózati erőforrásokat egy hálózati szolgáltatásvirtualizálási infrastruktúrát nyújtó szolgáltatások piacán osztják majd ki több adminisztratív tartomány között. Az ügyfelek közé tartoznak majd az otthoni felhasználóknak szolgáltatásokat kínáló alkalmazásslaválatók, az üzleti szolgáltatásokat nyújtó vállalati alkalmazásslaválatók, stb. A piac másik oldalán a számítástechnikai infrastruktúraszaválatók és hálózatüzemeltetők lesznek. Tekintettel az online szolgáltatások gyors telepítésének lehetőségére, az erőforrások elosztása és a szolgáltatásnyújtás dinamikus és rugalmas lesz. Ezt a virtualizációs technikák és az SDN alkalmazása biztosítja.

A következőkben modelleket és azokon végzett elemzéseket mutatok be a közvetítői szolgáltatások saválatátók közötti (B2B) árazásához, valamint a felhő- és hálózati szolgáltatások végfelhasználók számára történő (B2C) árazásához Stackelberg-játékok [12] alapján.

1. Téziscsoport. *Kutatásom során azt vizsgáltam, hogy a felhő-központú szolgáltatásnyújtás hogyan befolyásolhatja az Internet szerkezetét, és mely tényezők befolyásolhatják a saválatátókat a szolgáltatások közvetítésének árazásában. A modellt egy hálózatformáló játékként formalizáltam, és ennek egyensúlyi feltételeiről analitikai eredményeket vezettem le. Ezen a piacon továbbá modelleztem az ügyfelek és a saválatátók árval kapcsolatos döntéseit is: bemutatam hogyan formalizálják igényeiket az ügyfelek és hogyan választják ki a megfelelő erőforrásallokációt a kínálatból. A saválatátók szempontjából azt modelleztem hogyan viszonyulnak egymáshoz és levezettem hogyan kell meghatározniuk végáraikat a várható ügyféligenyekre való tekintettel.*

Megadtam azokat a közvetítői árakat, amelyek biztosítják a *status quo* fenntartását egy többszintű topológiában.

1.1. Tézis. [C1] *A G^0 gráf tartalmazzon számos, teljes hálóban összekapcsolt Tier-1 csomópontot, és minden Tier-1 csomópont alatt egy fa részgráfot, amelyben a közbenső csomópontoknak legalább k gyermekük van, és minden levélcsoMópont azonos t mélységben van. Továbbá bármely levélcsoMópont-pár m üzletet cserél; illetve a közvetítő csomópontok nem szerepelnek saválatátóként vagy vevőként. Legyen α egy üzleti link költsége. Ha minden saválatátó $\frac{\alpha}{k^{2t-3m}}$ alatt tartja a közvetítői árat, akkor a G^0 topológia egyensúlyban van.*

A kezdeti topológia szigorúbb feltételezése mellett a Tier-1 és Tier-2 egyensúlyi árakat egy 3-szintű topológiában vezettem le.

1.2. Tézis. [J1] *Tételezzük fel a saválatátók 3-szintű topológiáját, kezdetben a Tier-1 teljes hálón kívül bármiféle egyéb tranzit/társkapcsolat nélkül. To-*

	Erőforrások			Erőforrásigény	
	h. pontok	hálózatok	adatközpontok	kimenetel	val. változó
	\mathcal{A}	\mathcal{N}	\mathcal{D}	s	S
Kapacitás	∞	c_n	c_d	c_s	C
Késleltetés	0	l_n	l_d	l_s	L
Ár/költségvetés	0	p_n	p_d	b_s	B

1. táblázat. Az alkalmazott jelölések összefoglalása.

vábbá feltételezzük, hogy minden Tier-2 és Tier-3 szolgáltató egy-egy tranzit kapcsolatot építhet fel egy Tier-1-hez és egy Tier-2-hez költség nélkül.

Egyensúlyi állapotban a Tier-1 szolgáltatók mindegyike $\beta_1^* = \frac{n_1(n_1-1)\alpha}{2\gamma m}$ értékre állítja a közvetítői árat, ahol n_1 a Tier-1 szolgáltatók száma, γ a Tier-1 szintet elérő üzleti hányad, az m pedig a M üzleti mátrix elemeinek összege.

Egyensúlyban a Tier-2 szolgáltatók közvetítői ára

$$\beta_2^* = \frac{\alpha}{2\delta m} \left(n_1(n_1 - 1) + n_2(n_2 - 1) \int_{\frac{2\gamma m}{n_1(n_1-1)}}^{\infty} f(\mu) d\mu \right),$$

ahol δ a Tier-2-t elérő üzleti hányad, n_2 a Tier-2 szolgáltatók száma, $f(\mu)$ és $g(\mu)$ pedig a Tier-2 és Tier-3 szolgáltatópárok közötti üzleti tevékenység.

A felhő- és hálózati szolgáltatások végfelhasználói piaca felé fordulva bebizonyítottam, hogy egy általános topológiában az erőforrásfoglalás nehéz probléma az ügyfelek számára a különböző technológiai és gazdasági követelmények miatt, pl. kapacitás, késleltetés, költségvetés (Definíció 4.1 és 4.2).

Jelölje a \mathcal{A} a hozzáférési pontok halmazát, a \mathcal{N} a hálózati szolgáltatók halmazát és a \mathcal{D} az adatközpontok halmazát.

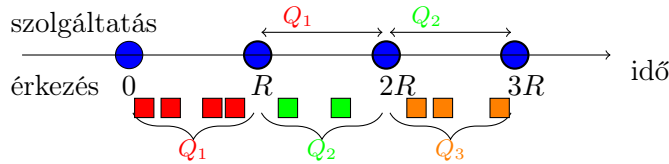
Modellemben az ügyfelek véletlenszerű erőforrásigényekkel jelentkeznek, amelyeket sikeres erőforrásfoglalás esetén a saját végfelhasználói számára nyújtandó szolgáltatás telepítésére használnak fel. A kívánt erőforrások jellemzői közé tartoznak az adott szolgáltatás-hozzáférési pontok, egy bizonyos mennyiségű számítási és hálózati kapacitás, a végpontok közötti maximális késleltetés és a maximális összköltség. Formálisan a 1. táblázatban összegzett jelöléseket használva:

4.1. Definíció (Erőforrásigény). Az erőforráskérést 4 elem $s = (a_s, c_s, l_s, b_s)$ határozza meg, amelyek

1. egy kiválasztott hozzáférési pontot $a_s \in \mathcal{A}$,
2. egy szükséges erőforrás méretet c_s (kapacitás),
3. egy, a végpontok (a hozzáférési pont és a választott adatközpont) közötti hálózati késleltetésre adott felső határt l_s ,

4. egy b_s költségkeretet (az ügyfél által fizetni hajlandó teljes költség felső határa).

írnak le.



1. ábra. Erőforrás-allokációs és szolgáltatásnyújtási időszakok: az egyes időszakokban (amelyek fix R hosszúságúak) kiszolgált kérések azok, amelyek az előző időszakban érkeztek.

Az erőforrás-allokáció és a szolgáltatásnyújtás időtartamát és azok kapcsolatát a 1. ábra szemlélteti.

4.2. Definíció (Erőforrás-elosztási és szolgáltatási időszakok). Az ügyfelek által bérelt erőforrások meghatározott időtartamra vannak allokálva; ez a szolgáltatásnyújtás időszaka. Ezen időszak előtt az ügyfelek egymás után érkeznek szolgáltatási kérésekkel; ezt nevezzük erőforrás-allokációs időszaknak. Az ügyfelek a kiválasztott erőforrásszolgáltatók erőforrás-allokációs időszakban adott árain foglalják le az erőforrásokat a szolgáltatásnyújtási időszakra.

1.3. Tézis. *[C2], [J2] A szolgáltatók összeköttetéseit reprezentáló hálózatban egy legfeljebb b_s költségű (vagy a legolcsóbb) megfelelő folyam megkeresése NP-nehéz, ha a hálózat tetszőleges gráf.*

Mivel az ügyfelek NP-nehéz problémával szembesülnek a feltételeknek megfelelő folyam kiválasztásával kapcsolatban, a bevezetett Stackelberg játékot, ahol a szolgáltatók vezetők, az ügyfelek pedig követők, egy sztochasztikus játékra redukáltam a szolgáltatók, mint játékosok között. Megmutattam, hogy a felhő- és hálózati erőforrások dinamikus árazása minden esetben idomul a piaci viszonyokhoz, és levezettem különböző, véges és végtelen kapacitású topológia-beállítások egyensúlyi árait (Lemma 4.1, 4.2, 4.3, 4.4, 4.5, 4.6).

4.1. Feltételezés. Az erőforrás-allokációs időszak alatti kérések száma a q paraméterrel rendelkező geometriai eloszlást követi, azaz minden kérés q valószínűséggel az utolsó a sorban, és legalább egy másik kérés következik $1 - q$ valószínűséggel.

4.1. Lemma. *Ha a 4.1. feltételezés teljesül, akkor a játék értéke:*

$$V^1(c, p) = \frac{\mathbb{P}(Z)(p\mathbb{E}(C|Z) + (1 - q)\mathbb{E}(W^1(c - C)|Z))}{q + \mathbb{P}(Z)(1 - q)},$$

ahol $W^1(c)$ a játék optimális értéke egy c kapacitású játékos számára, $\mathbb{P}(X)$ az X véletlen esemény valószínűsége, $\mathbb{E}(Y)$ egy Y valószínűségi változó várható értékét jelöli, és $Z \stackrel{\text{def}}{=} (B \geq pC)$, azaz a $B \geq pC$ esemény.

4.2. Feltételezés. A szolgáltatási kérelmek kapacitásértékei fixek $C \equiv 1$, az ügyfelek költségvetései egyenletes eloszlást ($U(\min, \max)$) követnek, az általánosság elvesztése nélkül $B \sim U(0, 1)$.



2. ábra. Egyetlen adatközpont esete.

4.2. Lemma. Ha a 4.1. és 4.2. feltételezések teljesülnek, akkor az 2. ábrán bemutatott beállításban az optimális ár a következő:

$$p^* = \frac{1 - \sqrt{1 - (1 - q)(1 - (1 - q)qW^1(c - 1))}}{1 - q}.$$

4.3. Lemma. Ha a 4.2. feltételezés teljesül és a szolgáltató kapacitása végtelen, azaz $c = \infty$, akkor az egyensúlyi ár $\frac{1}{2}$.

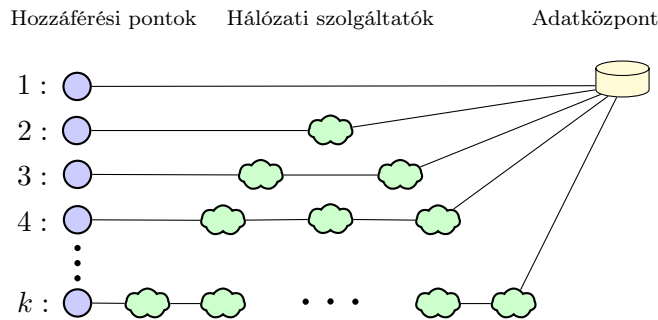
4.4. Lemma. Ha a 4.1. és 4.2. feltételezések teljesülnek, a hálózati szolgáltatók száma k és soros beállításban helyezkednek el, akkor egy Nash-egyensúlyt a következő egyenlet ír le: $0 = k^2(1 - q)p^{*2} - p^*(2k - q(k - 1)) + 1 - q(1 - q)W^s(c - 1)$.

4.5. Lemma. Ha a 4.2. feltételezés teljesül és a szolgáltatók kapacitása végtelen, azaz $c_i = \infty$, akkor a k , sorban elhelyezkedő csomópont esetén az egyensúlyi ár és az „anarchia ára” (angolul Price of Anarchy, PoA):

$$p^* = \frac{1}{k + 1}, \quad \text{PoA} = \frac{(k + 1)^2}{4k}.$$

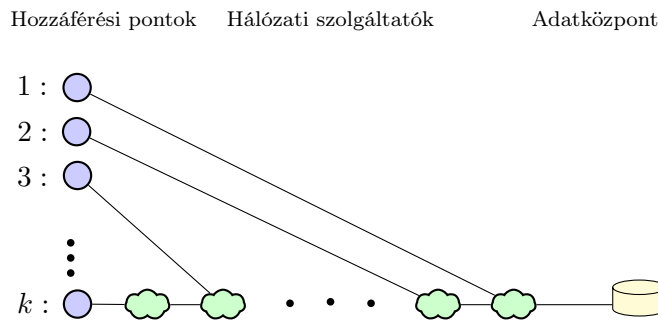
4.6. Lemma. Ha a 4.1. és 4.2. feltételezések teljesülnek, és két szolgáltató érhető el párhuzamosan, akkor a következő egyenletek megoldása adja meg a két játékos, i -vel és $-i$ -vel jelölve, egyensúlyi árait:

$$\begin{aligned} 0 = & p_i^{*2}(1 - q) - p_i^*(4 - 2p_{-i}^*(1 - q)) + 2 - (1 - q)(2W^p(c_i - 1, c_{-i}) + p_{-i}) \\ & + (1 - q)^2(p_{-i}(W^p(c_i - 1, c_{-i}) - W^p(c_i, c_{-i} - 1)) \\ & + (1 - q)^2(W^p(c_i - 1, c_{-i}) + W^p(c_i, c_{-i} - 1)). \quad (1) \end{aligned}$$



3. ábra. Párhuzamos utak soros hálózatokkal.

1.4. Tézis. [J2] Kifejeztem az egyensúlyi felhőszolgáltatási árakat egyszerű topológiákra (amelyekben egy vagy két adatközpont érhető el az ügyfél által közvetlenül vagy egy sor hálózati szolgáltatón keresztül) olyan esetekben, amikor az erőforrás-allokációs periódus során a kérések száma geometriai eloszlást követ.



4. ábra. Fenyőfa topológia.

Az eredményeket általánosítva eltekintettem a szolgáltatásigények kapacitás összetevőjétől, és levezettem a mesterséges, de bonyolultabb topológiákban (ahol sorban elhelyezkedő hálózati szolgáltatók párhuzamos és egymást átfedő utakat képeznek) előálló egyensúlyi árakat (3. és 4. ábrák, 4.7 és 4.8 lemmák). Az eredmények alapján formalizáltam a végtelen kapacitású felhő- és hálózati szolgáltatók egyensúlyi árait.

4.3. Feltételezés. A hálózati és felhőszolgáltatók elegendő kapacitással rendelkeznek a szolgáltatási kérelmek fogadásához: egy adott kérés elérhető útvonalainak halmazát nem korlátozzák kapacitáskorlátok.

4.7. Lemma. Ha a 4.2. és 4.3. feltételezések érvényesek, és ha a 3. ábrán látható topológia írja le a rendszert, és ha a kérésekben egyenletesen eloszlást követve jelentkezik a k hozzáférési pont, akkor az egyensúlyi árak a követke-

zők:

$$p_D^* = \frac{H_k}{k + H_k}, p_i^* = \frac{k/i}{k + H_k}, PoA \sim \frac{k + H_k}{4H_k}$$

ahol p_D^* az adatközpont ára, p_i^* az $i - 1$ hálózati szolgáltatót tartalmazó úton levő csomópontok árai és $H_k = \sum 1/j \sim \ln k$.

4.8. Lemma. *Ha a topológia megegyezik a 4. ábrán láthatóval, és a kérések egyenletesen érkeznek a k hozzáférési pontból, akkor a Nash-egyensúlyi ár:*

$$p_i^* = \frac{e_i(i+1)(i+2)\dots k}{(e_i+i)(e_{i+1}+i+1)\dots(e_k+k)},$$

ahol $e_1 = 1$ és $e_{i+1} = \frac{i+(i+1)e_i}{i+e_i}$. A hálózati szolgáltatók az általuk csatlakoztatott hozzáférési pontokhoz rendelt számok alapján vannak indexálva a 4. ábrán látható módon.

1.5. Tézis. [J2] Jelölje t a kért szolgáltatási utat, a \bar{t} pedig a t -n lévő csúcsok halmazát, kivéve a szolgáltatás hozzáférési pontot. A következő egyenletek megadják az összes szolgáltató $x \in G$ által támasztott árat:

$$0 = \sum_{s \in S} \sum_{x \in t, t \in \mathcal{T}_s} \mathbb{P}(S = s, T_s = t) \left[\left(1 - F_{B_s} \left(\sum_{y \in \bar{t}} p_y \right) \right) - f_{B_s} \left(\sum_{y \in \bar{t}} p_y \right) p_x \right], \quad (2)$$

ahol \mathcal{T}_s az s ügyféltől egy adatközpontba vezető útvonalak halmaza, p_x az x erőforrás-szolgáltató egységára, F_{B_s} és f_{B_s} az s ügyfél költségvetésének kumulatív eloszlása, illetve valószínűségi sűrűségfüggvénye.

A vevői igények előre becsült eloszlása mellett a szolgáltatói erőforrások egymáshoz viszonyított elhelyezkedése is kiemelten fontos a jövedelemmaximalizáló árak meghatározásában. Általános megfigyelésként kimutattam, hogy azok az adatközpont-szolgáltatók, amelyek közelebb helyezkednek el az ügyfelekhez (a köztük lévő hálózati szolgáltatók számát tekintve), magasabb árat tudnak megállapítani, és minél közelebb van egy hálózatszolgáltató az adatközpontokhoz, annál magasabb az általa diktált egyensúlyi ár.

4.2. Felhőn belüli erőforráskezelési és vezénylési módszerek

A felhőalapú számítástechnika az online szolgáltatások *de facto* szabványává nőtte ki magát, mivel olcsó és egyszerű megoldásokat kínál az erőforrás- és alkalmazáskezeléshez. Az elosztott infrastruktúra, például a perem- és köd-számítástechnika, minden tekintetben jelentős előrelépés az online szolgáltatások kínálatának költséghatékony módon történő bővítése felé.

A 4.2. fejezetben felhőkezelési technikákat és módszereket javasoltam az erőforrás-szervezési kihívások széles skálájára, például a megbízhatóságra, a skálázhatóságra, a késleltetéskezelésre és a költséghatékony működésre.

2. Téziscsoport. *Javasoltam egy rendszert és benne működő módszereket, amelyek magas megbízhatóságot és alacsony késleltetést biztosítanak gazdaságos módon nagyléptékű peremfelhőkben. Egy online és egy offline erőforrás-ütemező, és egy peremfelhő csomópont-szegmentáló eljárás kerül bemutatásra, amelyek együttesen nagy léptékben képesek kezelni a szolgáltatástelepítési kéréseket egy földrajzilag kiterjedt felhő infrastruktúrában. Javasoltam továbbá egy gépi tanulás-alapú automatikus skálázási módszert annak érdekében, hogy kezelni tudjam az online alkalmazások használati intenzitásának időbeli változékonyságát. Elemeztem a felhőalapú alkalmazások különböző telepítési lehetőségeinek hatását a válaszidőkben jelentkező késleltetésre és a memória-fogyasztásra, és javasoltam egy költséghatékony módszert mikroszolgáltatások futtatására.*

A késleltetés- és működéskritikus alkalmazások méretezhető és gazdaságos peremfelhő ütemezéséhez olyan rendszert javasoltam, amely nagy megbízhatóságot biztosít az alkalmazások számára azáltal, hogy biztonsági mentési erőforrásokat biztosít a peremfelhő csomópontjain. Ezeket helyőrzőknek neveztem el. Beállítottam a helyőrzők kapacitását az egy csomóponti meghibásodásokat orvosolni képes esetre (azaz bármely csomóponton az összes Pod egyszerre meghibásodhat). A helyőrzők kiszámítására és a Pod-ok elhelyezésére javasolt algoritmusok gyorsnak bizonyultak.

2.1. Tézis. *[C3, C4], [J3]. Megmutattam, hogy az online ütemezés, az offline újraütemezés és az infrastruktúra csomópontszegmentálási módszerek polinomiális komplexitásúak.*

A 4.9., 4.10., 4.11., 4.12., 4.13., 4.14. és 4.15. lemmákra alapozva kiszámoltam a heurisztikus online ütemező algoritmus válaszána minőségét a feladatátvételtre szánt erőforrások mennyisége, azaz a helyőrzők teljes összege szempontjából.

4.3. Definíció (Rügy). A gráf csomópontja akkor rügy, ha legalább egy levélhez kapcsolódik.

4.9. Lemma. *Adott számú csomópont ($|V| = n$) és átmérő (d) esetén a rügyek számának növekedésével (b) az optimális megoldás (OPT) monoton nő.*

4.10. Lemma. *Adott számú csomópont ($|V| = n$) és adott számú rügy (b) esetén a gráf növekvő átmérőjével d a heurisztikus algoritmusunk által megadott helyőrzők száma ($HEUR$) monoton növekszik.*

4.11. Lemma. $OPT \geq \frac{d+1}{3}$

4.12. Lemma. $OPT \geq b$

4.13. Lemma. $HEUR \leq n - 1$.

4.14. Lemma. *Adott átmérőjű egyszerű, összefüggő gráfon a rügyszek számát a következő szabályok adják.*

1. Ha $d = 1$, akkor $b = 0$;
2. Ha $d = 2$, akkor $0 \leq b \leq 1$;
3. Ha $3 \leq d \leq \frac{n}{2}$, akkor $0 \leq b \leq \frac{n}{2}$;
4. Ha $d = \frac{n}{2} + k, k > 0 (\frac{n}{2} + 1 \leq d \leq n - 3)$, akkor $0 \leq b \leq (\frac{n}{2}) - k + 1$;
5. Ha $d = n - 2$, akkor $1 \leq b \leq 3$;
6. Ha $d = n - 1$, akkor $b = 2$.

4.15. Lemma. *A heurisztikus és az optimális megoldás közötti közelítés aránya adott maximális átmérőhöz és minimális rügyszámhoz:*

1. ha $d = 1$ és $b = 0$, then $HEUR = OPT = 2$;
2. ha $d = n - 3$ és $b = 0$, akkor $HEUR \leq 3OPT$;
3. ha $d = n - 2$ és $b = 1$, akkor $HEUR \leq 3OPT$;
4. ha $d = n - 1$ és $b = 2$, akkor $HEUR \leq 3OPT$.

2.2. Tézis. [J3] *A javasolt online ütemezési módszer egy 3-approximációs algoritmus a Pod-ok helyőrzőinek együttes elhelyezésére ($HEUR \leq 3OPT$).*

A Kubernetes [13], a széles körben használt felhőkezelő platform, beépített megoldással rendelkezik a platformon futtatott alkalmazások méretezéséhez. A Horizontal Pod Autoscaler (HPA) nevű beépített eljárás működését számos paraméter befolyásolja. Vannak felhőszintű beállítások, például a lefelé skálázás stabilizálása, a Pod szinkronizálás periódusa és a skálázási tolerancia, valamint vannak HPA szintű paraméterek, mint például a minimális és maximális pods szám, a tetszőleges metrikák szerinti skálázási küszöb. Alapértelmezés szerint a HPA a processzor (Central Processing Unit (CPU)) kihasználtságon alapul: a HPA meghatározott időszakonként lekéri a CPU használati adatokat a rendszerből, és eldönti, hány Poddal kell rendelkeznie az alkalmazásnak. Az automatikus skálázási módszer teljesítményének elemzéséhez veszteséges és veszteségmentes modelleket építettem, hogy minél pontosabban leírják a HPA viselkedését.

2.3. Tézis. [C5], [J4] *Analitikai modelleket javasoltam a Kubernetes automatikus skálázó eljárásának leírására.*

A veszteségmentes modell azt feltételezi, hogy az alkalmazás terhelése és a CPU használat közötti összefüggés állandó, és hogy a kiszolgálási idők exponenciális eloszlást követnek. A kiszolgálási rátát μ -vel jelölöm. A

javasolt modell a kérések érkezési folyamatát Markov-modulált Poisson folyamatként (MMPP) írja le, m állapottal; az állapotátmeneti mátrixot Q jelöli, az m érkezési rátát pedig $\lambda_1, \lambda_2 \dots, \lambda_m$. Ahogy a HPA periodikusan állítja a szerverek számát, hasonló módon a modell is időnként dinamikusan igazítja a kiszolgálók számát az érkezési ráta változásának megfelelően. A HPA reaktív, azaz az érkezési arány változása a következő skálázási intervallumban érvényesül a szerverek számára. Ezért a szerverek száma a modellben is követi az MMPP érkezéseket, de egy skálázási periódussal késik. Legyen $\lambda(t)$ az érkezési ráta t időpontban. Feltételezve, hogy az MMPP egy adott állapotban eltöltött ideje (pl. perc) nagyságrendekkel hosszabb, mint a kérések beérkezése közötti idő (pl. ezredmásodperc), a szerverek száma t -ben kiszámítható a következő képlettel:

$$C(t) = \left\lceil \frac{\lambda(t-1)}{\rho\mu} \right\rceil, \quad (3)$$

ahol ρ az M/M/c sormodellekben a kiszolgálók kihasználtsága és a következőképpen számítható ki: $\rho = \lambda/(c\mu)$, λ érkezési rátával, c kiszolgálóval és μ kiszolgálási rátával. A kiszolgálók kihasználtsága megegyezik az HPA kezelési paraméterével.

4.4. Definíció (Alkalmazási profil). Az alkalmazásprofil egy $AP: \mathbb{N}^+ \rightarrow \mathbb{R}^+$ függvény, amely a Podonkénti kiszolgáltatási rátát az aktív Podok számához rendeli.

A javasolt veszteséges modell egy összetettebb, diszkrét idejű sorbanálási modell, amely a HPA c_{i+1}^* döntését, azaz a következő skálázási periódusra tervezett Podok számának kiszámítását utánozza. A kiszolgált kérések száma és a CPU használat szorosan összefügg, és ez utóbbi a HPA skálázási döntéseinek alapja. Legyen a kérések száma a rendszerben az i időszakban

$$L_i = L_{i-1} - M_{i-1} + \Lambda_i - T_i \quad (4)$$

ahol Λ_i a beérkezett kérések száma, M_i a kiszolgált kérések száma az i időszakban, és T_i a sorban álló kérések száma, amelyek a i időszakban járnak le. M_i definíciója a következő:

$$M_i = \min\{c_i AP(c_i), L_i\} \quad (5)$$

ahol c_i a Pod-ok száma, a $AP(c_i)$ pedig a 4.4. definícióban leírt alkalmazásprofil által adott kiszolgálási ráta, azaz M_i a rendszerben lévő kérések számának és azon kérések számának a minimuma, amelyeket a rendszer ki tud szolgálni a i időszakban. Az L_i tartalmazza az i időszakban érkezett kéréseket, de nem tartalmazza az $i-1$ időszakban kiszolgált kéréseket, sem az i időszakban elvesztett kéréseket. A kezdeti periódusban üres sor feltételezhető, azaz $i = 0$, $L_0 = 0$, $c_0 = 1$, $\Lambda_0 = 0$.

A numerikus kiértékelések alapján a modellek elfogadhatónak bizonyultak az alapértelmezett automatikus skálázó teljesítményét reprezentáló szimulációs összehasonlítások céljára. Ezért ezeket a modelleket használtam összehasonlítóképpen a javasolt gépi tanulás alapú automatikus skálázási módszerem értékeléséhez. Fontosnak tartottam működésükben lényegesen eltérő gépi tanulási modellek kombinálását az alkalmazáshasználati előrejelzésekhez: az ensemble-alapú modellek jól ismert erősségei mellett a numerikus eredmények azt mutatták, hogy nem létezik egyetlen olyan gépi tanulási módszer, amely mindig jól közelítené az optimális skálázást, mivel a bemeneti kérések általában változó dinamikát mutatnak a nap folyamán, ezért a különböző módszerek epizodikusan jól vagy rosszul teljesítettek.

2.4. Tézis. [C5], [J4] *Javasoltam egy gépi tanulás alapú ensemble modellt és az arra épülő prediktív automatikus skálázási módszert felhőben telepített alkalmazásokhoz. Bemutattam, hogy a javasolt módszer jelentős költségmegtakarítást ér el (akár 50%) az alapértelmezett benchmarkhoz képest, főként az elveszett kérések jelentős csökkenésének köszönhetően.*

A skálázás költségét nagymértékben meghatározza az alkalmazás skálázási egységekre való szervezése. Egyrészt az alkalmazás-összetevők közös skálázási egységben való elhelyezése alacsonyabb működési késleltetéseket eredményez, így jobb QoS biztosítható az alkalmazás felhasználói számára. Másrészt a kisebb modularitás felesleges erőforrás-felhasználást eredményez a nagyléptékű működési időszakok során. A 4.16. és 4.17. lemmák alapján javaslatot tettem ez utóbbi költség minimalizálására.

Az alkalmazást alkalmazás-részek, úgynevezett modulok, véges halmazaként modelleztem, amelyek külön skálázási egységekbe csoportosíthatók. Feltételezem, hogy e modulok bármilyen kombinációja összecsomagolható. A modulokat egy adott időszakon belüli átlagos memóriafogyasztásuk alapján modellezem. Ezeket az értékeket r_1, r_2, \dots, r_n jelöli. Mindegyik modulhoz tartozik még egy további s érték, amely az idővel súlyozott átlagos skálázási tényezőjét tükrözi.

4.16. Lemma. *Tetszőleges számú skálázási egység esetén a skálázási költség úgy minimalizálható, hogy a modulok skálázási egységekbe való csoportosítása a skálázási ráták sorrendjét követi.*

4.17. Lemma. *Tetszőleges számú (x), skálázási rátájuk szerint növekvő sorrendbe vett, minimális skálázási költséggel rendelkező skálázási egység esetén*

$$s_b \geq \frac{s_j^g r_b + s_{j-1}^g \sum_{i \in j-1} r_i}{r_b + \sum_{i \in j-1} r_i} \quad \forall j < x \quad (6)$$

teljesül, ahol a b modulok a skálázási egységek határain vannak, és a skálázási rátáikat s_b jelöli.

2.5. Tézis. [J5] *Analitikai modellt javasoltam a nagy léptékű alkalmazásfutási időszakok alatti erőforrás-többlet és az alkalmazás több skálázási egységbe rendezéséből fakadó késleltetési többlet közötti döntési helyzet leírására. Jelölje ρ az erőforrásigényt, és legyen a skálázási tényező monoton növekvő, folytonos és minden pontban differenciálható függvénye $\sigma(\rho)$. Ekkor*

$$\frac{d\sigma}{d\rho}\rho_L + \sigma - \sigma_R = 0 \quad (7)$$

teljesül a skálázási csoportok határaitra (szomszédos skálázási csoportok határai, ha a skálázási tényezők szerint vannak rendezve a kumulatív erőforrásigény dimenziója mentén) amikor az erőforrás-többlet minimális. A ρ_L a bal oldali skálázási csoport szélességét jelöli, a σ a bal oldali skálázási csoporthoz tartozó skálázási ráta értéke, a σ_R pedig a jobb oldali skálázási csoport skálázási rátáját jelöli ebben az elrendezésben.

Ennek következtében megállapítható, hogy viszonylag nagy ugrások vannak a skálázási csoportok skálázási rátái között egy olyan beállításban, amely a skálázási költségre van optimalizálva.

4.3. A felhőhozzáférés erőforrás-kiosztása

A vezeték nélküli hozzáférési hálózatokban biztosítható QoS a távközléssel kapcsolatos kutatások fontos tényezője volt, és a hálózatosodó társadalom elterjedésével az is lesz. Mivel ezek a hálózatok általában több ügyfél között vannak megosztva, a hálózati erőforrások biztosítása kiemelkedően fontos az ügyfelek által a felhőalkalmazásokhoz való hozzáférés során észlelt QoS szempontjából. Ebben a részben ezért két szolgáltatásminőség-biztosítási keretrendszert mutatok be, amelyekkel a hálózatüzemeltetők kezében lévő módszereket felhasználóvezérelt vagy felhő-alapú minőségbiztosítás lehetőségével bővítem.

3. Téziscsoport. *Javasoltam egy koordinációmentes és egy koordinált erőforrás-allokációs sémát a hálózati sávszélesség felosztására felhőalkalmazások ugyanazon hozzáférési hálózaton lévő kliensei között.*

Először az árverésként megvalósított koordinációmentes erőforrás-kiosztásra definiáltam egy modellt, ahol a kliensek rövid időszakokra többlet sávszélességért licitálhatnak a hálózat üzemeltetőjénél történő ajánlattétellel. A szolgáltató az erőforrásokat a következő időszavokra tett ajánlatok alapján osztja ki a felhasználóknak.

A rendszer a következőképpen viselkedik. Minden egyes időrészben a rendszer az \mathcal{S} állapotok egyikében van, amely leírja a felhasználók *sávszélességigényeit* és a *fennmaradó költségvetéseiket*. Ha kevesebb igény van, mint amennyit a rendszer egy időrészben kezelni képes, akkor természetesen minden kérés elfogadásra kerül. Azonban amikor az erőforrásigény meghaladja

a rendelkezésre álló kapacitást, vagyis több kiszolgálásra váró munka van a rendszerben, mint amennyit ki lehet szolgálni, egy adott munka vagy elfogadásra kerül, vagy kiszolgáltatlan marad az időérés végére.

A Markov döntési folyamaton (MDP) alapuló központi optimalizálási megközelítés minden állapotban megköveteli a lehetséges műveletek meghatározását, azaz $\pi_i(b_i|s) = \mathbb{P}[A_i^t = b_i|S_t = s] \forall t$, ahol A_i^t jelöli azt a valószínűségi változót, amely az i felhasználó tevékenységét írja le a t időrásben, amikor a rendszer s állapotban van. A rendszer leírásának nagy memória-igénye miatt az MDP alapú elemzés nem kivitelezhető. Az állapottér alapú optimalizálás helyett a problémát egy licit alapú sztochasztikus játékként kezelem.

3.1. Tézis. [C6] *Vickrey típusú árverésen alapuló, koordinálatlan erőforrás-allokációs modellt javasoltam felhőalkalmazások elérésére, és numerikus elemzéssel olyan heurisztikus stratégiákat mutattam be, amelyek bizonyos forgalmi terhelési esetekben sikerre vezetnek.*

Az eredmények azt mutatják, hogy az *előzékeny* ajánlattételi stratégia hosszú távon kifizetődő ha az átlagos rendszerterhelés alacsony, mivel a legtöbb esetben a viszonylag alacsony ajánlat elegendő a nyereséshoz az árverésen, egyúttal átmenekíti a költségkeret nagy részét a túlterhelt időszakokra. Amikor a rendszer közel van a telítettségéhez, érdemes a munkák tényleges értékével licitálni (*racionalis stratégia*), hogy megelőzzük az egyéb ajánlattételi stratégiákat alkalmazó felhasználókat.

A következő lépésben egy központosított erőforrás-allokációs sémában javasoltam az erőforrás-hatékonyság biztosítását azáltal, hogy lehetővé tettem a felhő számára, hogy szabályozza a felhőbe feldolgozásra küldött adatfolyamok sebességét. A példaként tekintett videófolyam-feldolgozó IoT rendszerben a felhőben futó optimalizáló szoftver képes előrejelezni a rögzíteni kívánt jövőbeni események lehetséges helyét és időpontját több videófolyamban. Az előrejelzés alapján egyes videófolyamok nagyobb sávszélességet kapnak, más kamerák feed-jeit pedig kikapcsoljuk, vagy alacsonyabb sávszélesség-beállításokra állítjuk át. Így a megosztott feltöltési sávszélesség többnyire az alkalmazás szempontjából fontos forgalmi adatok által van elfoglalva.

3.2. Tézis. [C7] *Egy olyan koordinált erőforrás-allokációs sémát javasoltam a felhőalkalmazásokhoz való hozzáférés megosztásához, amely képes előrejelezni az adatátviteli igényeket és a kiosztott hálózati erőforrás mennyiségének hatását a QoS-re. Egy dinamikus programozáson alapuló megoldást javasoltam a QoS maximalizálási problémára, és bebizonyítottam, hogy az optimum $O(nmB_{max}^{m+1})$ lépésben kiszámítható, ahol n az adatforrások számát, m a különböző típusú erőforrások, például sávszélesség, CPU a felhőben, stb., számát, a $\{B_1, \dots, B_m\}$ pedig a megfelelő források rendelkezésre álló mennyiségét jelzi, és $B_{max} = \max\{B_1, \dots, B_m\}$.*

Megjegyzendő, hogy a javasolt módszer alkalmazásához minden adatforrás és minden típusú erőforrás esetében a különböző kiosztási állapotok számának végesnek kell lennie. Az állapotok tényleges száma attól függ, hogy hány különböző értéket vehet fel egy adott lefoglalt erőforrászelet. Abban az esetben, ha a kapott állapottér viszonylag kicsi, mert pl. a rendelkezésre álló teljes sáv szélességet nagy egységekben osztjuk fel az adatforrások között, akkor az algoritmus futásideje csökkenthető.

5. Az eredmények hasznosíthatósága és hatása

5.1. Szolgáltatók közötti árképzés

A 4.1. fejezetben bemutatott első modell három kutatási terület metszetébe tartozik: az *5G felhő- és hálózati infrastruktúra-szolgáltatók árképzési sémájával* foglalkozom, amely az üzleti kapcsolati hálójuk fejlődését indukálja, mindezt pedig a *hálózatképző játékok* módszertanával elemzem. Sejtésem szerint a jövőbeli 5G-hez kötődő üzleti háló elrendezése rendkívüli mértékben fog függeni a jelenlegi internetes topológiától. Ezért azt várom, hogy a *jelenlegi internetes tranzit és társkapcsolatok* által meghatározott többrétegű topológia lesz a kiindulási struktúra, amelyet persze az új 5G szolgáltatók és környezet tovább fog alakítani. A tanulmányom az üzleti kapcsolatok fenntartásának és az üzleti közvetítők alkalmazásának dilemmáját vizsgálja.

Az elemzés eredményeként kimutattam, hogy a magas szintű szolgáltatókra motiváló erők hatnak az irányba, hogy közvetítői költségeiket alacsonyan tartsák. Az a jelenség, ahogy megpróbálják menteni a *status quo*-t a szolgáltatók közötti üzleti kapcsolatok terén, általános konstruktív következményekkel jár az egész rendszerre nézve: a többszolgáltatós szolgáltatások létrehozásának közvetítői költsége felülről korlátos. Ezt a korlátot maga a topológia és egy új kapcsolat létrehozási költsége határozza meg. A bemutatott munka úttörő a többszolgáltatós heterogén erőforrás-szolgáltatók árképzésének elemzésében. Az eredmények gyakorlati lehetőségeket is rejtenek az NFV vezérlési megoldások és a szolgáltatói szövetségek struktúrájának kialakításában, mindkettő kulcsfontosságú az 5G jövőkép megvalósításában.

A jövőbeni online szolgáltatásokat tehát gazdasági okokból infrastruktúra-szolgáltatók szövetségei fogják biztosítani: az erőforrások hatékony felhasználása és az ügyfelek széles földrajzi elérése érdekében. A 4.1. fejezetben a második modellem az infrastruktúra-szolgáltatókat, mint az ügyfelekért egymással versengő, a profitmaximalizálást szem előtt tartó játékelméleti játékosként kezeli. Megmutattam, hogy az ügyfelek egy nehéz problémát kell, hogy megoldjanak, ha mind műszaki, mind üzleti igényeiket szeretnék kielégíteni, nem beszélve a költségek minimalizálásáról. Levezettem a szolgáltatók legjobb árképzési stratégiáit az ügyfelek előre megbecsült kereslete alapján. A legfőbb gyakorlati eredményként megmutattam, hogy az egyensúlyi árak és az elérhető bevétel erősen függ a szolgáltató hálózaton belüli elhelyezke-

désétől.

5.2. Felhőkezelési technikák

A 4.2. fejezetben javasolt felhőkezelési platform egy általános virtuális infrastruktúra ütemezőt egy földrajzilag elterjedt infrastruktúra kezelőjévé változtatja. A javasolt platform két ütemezési algoritmusra épül, amelyek kiválóan támogatják a késleltetés szempontjából kritikus alkalmazásokat, és egy szegmentálási algoritmusra építve könnyen megbirkóznak nagyméretű, földrajzilag kiterjedt infrastruktúrával is. A Kubernetes [13] a közelmúltban a legelterjedtebb használt felhőplatformmá vált. Elsősorban webalkalmazásokat futtató adatközpontok vezénylésére szolgál. Legfontosabb képességei, például az öngyógyítás és az automatikus méretezés, hatalmas fejlesztői közösséget vonzottak ennek a nyílt forráskódú projektnek, így az rohamosan fejlődik. A javasolt ütemező algoritmusokat beintegráltuk a Kubernetesbe, hogy alkalmassá tegyük peremfelhő infrastruktúra kezelésére. Mivel a késleltetésre érzékeny alkalmazásokat érdemes peremfelhőbe telepíteni, egy olyan topológiatudatos Kubernetesre változatra van szükség, amely hálózati késleltetés figyelembevételével terjeszti ki a széles körben használt szolgáltatás-készletét. Ezen túlmenően, mivel a peremfelhő-infrastruktúra hajlamos a meghibásodásokra és költségesnek tekinthető a karbantartása, az öngyógyító funkcióknak nagyobb hangsúlyt kell kapniuk, mint az alap Kubernetesben. A peremfelhő-ütemezőnk architektúráját és működését a rangos ACM Sigcomm konferencián [C4] mutattuk be, forráskódját pedig nyilvánosságra hoztuk.

A mikroszolgáltatás-alapú alkalmazások koncepciója és a hozzájuk tartozó technológiák új módot kínálnak az alkalmazásfejlesztésre és a szolgáltatásüzemeltetésre. A kulcsfontosságú felügyeleti funkciókat, mint például az erőforrás-skálázás, a mögöttes felhőplatformok biztosítják, de az újszerű szolgáltatásokat felhőplatformokon keresztül biztosító alkalmazásslátszóknak megfelelő konfigurációs felületekre van szükségük a működés felső szintű irányításához. Amint azt a 4.2. fejezetben bemutattuk, megvizsgáltuk a Kubernetes jelenleg elérhető automatikus skálázóját, majd megterveztünk egy új, proaktív skálázó motort, amely több gépi tanulás-alapú előrejelzési módszert alkalmaz a működés optimalizálása érdekében. A megvalósított, HPA+ névre keresztelt automatikus skálázót valós mérésekkel validáltuk, ahol a bemenő forgalmat egy egyetemi hálózatról gyűjtött forgalmi adatokból állítottuk össze. Azt találtuk, hogy az automatikus skálázó motorunk képes jelentősen csökkenteni az elutasított kérelmek számát némileg magasabb erőforrás-felhasználás árán, a többlet erőforrás-felhasználás azonban elenyésző a normál működéshez képest. Bár a HPA+ forráskódját nem hoztuk nyilvánosságra, a [C5], [J4] alapján az ensemble-alapú skálázási logika könnyen reprodukálható. Az eredményekkel kapcsolatban egy szabadalmi bejelentést [P1] is benyújtottam.

Továbbra is a mikroszolgáltatásokra koncentrálva bemutattam, hogy az

alkalmazástervezőnek át kell gondolnia a 4.2. fejezetben meghatározott költségek, azaz a skálázási és kommunikációs költségek, relatív fontosságát, hogy megtalálja a komponensek optimális elrendezését. Ahhoz, hogy ezeket a költségelemeket összehasonlíthatóvá tegyük, meg kell határozni az alkalmazásmódulok közötti hívási késleltetés okozta QoS csökkenés pénzbeli értékét, mivel a memóriefoglalás árának, azaz a skálázási költségnek van pénzben kifejezett ára. Ezután az alkalmazástervező alkalmazhatja az általunk javasolt optimalizált csoportosítást ahelyett, hogy i) csupán az alacsony válaszidőt szem előtt tartva a mikroszolgáltatási komponenseket egymáshoz közel helyezi el, vagy ii) egy teljesen elosztott elrendezést követ a legalacsonyabb skálázási költségért.

5.3. Felhő alatti erőforrás-megosztás

A 4.3. fejezetben bemutatott koordinációmentes hálózati sáv szélesség-kiosztással kapcsolatos modellünk kellően rugalmas ahhoz, hogy lefedjen különféle optimalizálási célokat, amelyek a hálózatüzemeltetők számára érdekesek lehetnek. A szolgáltató preferenciáinak érvényesítése a költségvetések kiosztásával érhető el, legyen az az átlagos késleltetés minimalizálása, a méltányosság maximalizálása vagy az kiéhezés elkerülése. A felhasználók azt a stratégiát választják majd, amely a legjobban megfelel a költségkeret korlátain belül.

Ami a koordinált sémát illeti, az IoT rendszerünk validálása érdekében egy labdabecsapódás-érzékelő szolgáltatást telepítettünk egy squashpályára. Az elülső falon történő pontos becsapódási hely és idő meghatározása, a játékosok pozíciói és az ütőutak mellett, fontos információt hordoz – labdasebesség, labda-röppálya, ütéstípus –, ami különböző felhasználási esetek alapjául szolgálhat, pl. videóbíró, edzéskiértékelés, pontszámlálás és kiterjesztett valóság-alapú játékok. Ezek az alkalmazások, hasonlóan az online játékokhoz, szigorú késleltetési korlátokat támasztanak a jó felhasználói élmény érdekében.

Adatokat gyűjtöttünk egy squash pályáról különböző *szenzorokkal*: az ütőérzékelők, mikrofonok és kamerák által szolgáltatott adatok egy kiválasztott részhalmazát a felhő alapú eseményészlelési folyamathoz irányítottuk. Az észlelt eseményeket beépítettük az előre definiált *adatmodellbe*, ezáltal frissítve a rendszer aktuális állapotát. Azután az *adatmodell* alapján előrejelzést adtunk a rendszer közeljövőbeni várható állapotáról. Végül a predikciók alapján minden (szimulált) squashpályához kiszámítottuk a hasznossági függvényeket, és az *erőforrásfoglaló* modul eldöntötte, hogy milyen módon ossza ki a sáv szélességet és a szervererőforrásokat a squashpályák között. További információk elérhetőek a projekt honlapján [14].

Az eredményekhez szorosan kapcsolódóan két szabadalmat jegyeztem be: [P2, P3].

A tézisekhez kötődő saját közlemények

Folyóiratcikkek

- [J1] Laszlo Toka, Akos Recse, Mate Cserep, and Robert Szabo. On the mediation price war of 5G providers. *Electronics*, 9(11), 2020.
- [J2] Laszlo Toka, Marton Zubor, Attila Korosi, George Darzanos, Ori Rotenstreich, and Balazs Sonkoly. Pricing games of NFV infrastructure providers. *Telecommunication Systems*, 76:219–232, 2020.
- [J3] Laszlo Toka. Ultra-reliable and low-latency computing in the edge with Kubernetes. *Journal of Grid Computing*, 19(3), 2021.
- [J4] Laszlo Toka, Gergely Dobreff, Balazs Fodor, and Balazs Sonkoly. Machine learning-based scaling management for Kubernetes edge clusters. *IEEE Transactions on Network and Service Management*, 18(1):958–972, 2021.
- [J5] Laszlo Toka. The shape of your cloud: How to design and run polyolithic cloud applications. *IEEE Access*, 10:97971–97982, 2022.

Konferencia cikkek

- [C1] Mate Cserep, Akos Recse, Robert Szabo, and Laszlo Toka. Business network formation among 5G providers. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018.
- [C2] Laszlo Toka, Janos Tapolcai, George Darzanos, and Balazs Sonkoly. On pricing of 5G services. In *IEEE Global Communications Conference (GLOBECOM)*, 2017.
- [C3] Laszlo Toka, David Haja, Attila Korosi, and Balazs Sonkoly. Resource provisioning for highly reliable and ultra-responsive edge applications. In *IEEE International Conference on Cloud Networking (CLOUDNET)*, 2019.
- [C4] David Haja, Mark Szalay, Balazs Sonkoly, Gergely Pongracz, and Laszlo Toka. Sharpening Kubernetes for the Edge. In *ACM SIGCOMM Conference Posters and Demos*, 2019.
- [C5] Laszlo Toka, Gergely Dobreff, Balazs Fodor, and Balazs Sonkoly. Adaptive AI-based auto-scaling for Kubernetes. In *IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, 2020.

- [C6] Laszlo Toka, Mark Szalay, David Haja, Geza Szabo, Sandor Racz, and Miklos Telek. To boost or not to boost: a stochastic game in wireless access networks. In *IEEE International Conference on Communications (ICC)*, 2020.
- [C7] Laszlo Toka, Balazs Lajtha, Eva Hosszu, Bence Formanek, Daniel Geberger, and Janos Tapolcai. A resource-aware and time-critical IoT framework. In *IEEE Conference on Computer Communications (INFOCOM)*, 2017.

Szabadalmak

- [P1] Marcial Gutierrez, Laszlo Toka, Balazs Fodor, and Balazs Sonkoly. Anomaly-aware cloud resource management system receiving external information, and including short- and long-term resource planning, 2022. EPO PCT pending P103161WO01, US pending P103161US01.
- [P2] Istvan Godor, Gabor Feher, Andras Balazs Lajtha, Laszlo Toka, and Attila Vidacs. System and method for resource-aware and time-critical IoT frameworks, 2021. US 11190409.
- [P3] Istvan Godor, Gabor Feher, Andras Balazs Lajtha, Laszlo Toka, and Attila Vidacs. Technique for online video-gaming with sports equipment, 2022. US 11318377.

Hivatkozások

- [1] Amazon Web Services (AWS): Elastic Container Service (ECS as CaaS) and Lambda (FaaS). <https://aws.amazon.com/>.
- [2] Google Cloud: Google Kubernetes Engine (GKE as CaaS) and Google Cloud Functions (FaaS). <https://cloud.google.com/>.
- [3] Microsoft Azure: Azure Kubernetes Service (AKS as CaaS) and Azure Functions (FaaS). <https://azure.microsoft.com/>.
- [4] Pal Varga, Jozsef Peto, Attila Franko, David Balla, David Haja, Ferenc Janky, Gabor Soos, Daniel Ficzer, Markosz Maliosz, and Laszlo Toka. 5G support for Industrial IoT Applications— Challenges, Solutions, and Research gaps. *Sensors*, 20(3):828, 2020.
- [5] Akhil Gupta and Rakesh Kumar Jha. A survey of 5G network: Architecture and emerging technologies. *IEEE Access*, 3:1206–1232, 2015.
- [6] Nisha Panwar, Shantanu Sharma, and Awadhesh Kumar Singh. A survey on 5g: The next generation of mobile communication. *Physical Communication*, 18(P2):64–84, 2016.

- [7] Jiafu Wan, Shenglong Tang, Zhaogang Shu, Di Li, Shiyong Wang, Muhammad Imran, and Athanasios V. Vasilakos. Software-defined industrial internet of things in the context of industry 4.0. *IEEE Sensors Journal*, 16(20):7373–7380, 2016.
- [8] ETSI. White Paper: Network Functions Virtualisation (NFV). http://portal.etsi.org/nfv/nfv_white_paper2.pdf, 2013.
- [9] ONF. Software-defined networking: The new norm for networks (white paper), 2012. <https://www.opennetworking.org>.
- [10] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, 2008.
- [11] Muhammad Alam, Joao Rufino, Joaquim Ferreira, Syed Hassan Ahmed, Nadir Shah, and Yuanfang Chen. Orchestration of microservices for iot using docker and edge computing. *IEEE Communications Magazine*, 56(9):118–123, 2018.
- [12] Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. The MIT Press, 1994.
- [13] Kubernetes: Production-Grade Container Orchestration. <https://kubernetes.io>.
- [14] Smartactive project. <https://sites.google.com/gsuite.tmit.bme.hu/smartactive/>.