



M Ű E G Y E T E M 1 7 8 2

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
AUTOMATIZÁLÁSI ÉS ALKALMAZOTT INFORMATIKAI TANSZÉK

MAROSI Attila Csaba

Önkéntes Számítási Rendszerek Kihívásai és Formális Aspektusai

Challenges and Formal Aspects of Volunteer Computing

Ph.D. Értekezés Tézisei

Témavezetők:

KACSUK Péter
MTA SZTAKI, LPDS

JUHÁSZ Sándor
BME, AUT

Budapest, 2016.



MTA Magyar Tudományos Akadémia
SZTAKI Számítástechnikai és Automatizálási Kutatóintézet

Tartalomjegyzék

1. Előzmények és célkitűzések	1
2. Kutatási módszertan	3
3. Új kutatási eredmények	4
4. Gyakorlati alkalmazás	10
Publikációs lista és irodalomjegyzék	11

1. Előzmények és célkitűzések

A Grid víziója [68] a földrajzilag elosztott számítási és háttértár erőforrások tökéletes és transzparens összekapcsolását célozta meg, amelyből egyszerűen lehet kapacitást igényelni és használni vagy saját kapacitását felajánlani, hasonlóan az elektromos-hálózathoz (innen származik a *Grid* név). A létrejött szolgáltatás-grid (Service Grid – SG) rendszerek a kapacitás felhasználását teszik lehetővé. Azonban a kapacitás felajánlása és karbantartása erőforrásigényes, így kevés számú üzemeltető jött létre, ellenben ezek nagyobb mennyiségű kapacitás felett rendelkeznek (tipikusan egyetemek és kutatóintézetek). Azonban a kapacitás igénylése és felhasználása sem egyszerűnek, megfelelő jogosultságokkal lehetséges (tipikusan egy hitelesítési folyamat után). Valamint a felhasználónak tisztában kell lennie a használt rendszer mögöttes működésével, hogy a gyakran felmerülő problémákat kezelni tudja. Ma már látszik, hogy a SG-ek nem teljesítik a Grid eredeti vízióját. Az Önkéntes Számítási Rendszerek (Volunteer Computing – VC) és Desktop Grid-ek (DG-ek) a szolgáltatás-grid-ekhez hasonlóan erőforrásokat gyűjtenek össze és tesznek elérhetővé felhasználásra. A SG-ekkel ellentétben azonban ezek nem dedikáltak, hanem a már használatban lévő, például irodai vagy önkéntesek által felajánlott – saját – számítógépek kihasználatlan kapacitásait jelentik. Azaz a DG/VC rendszer üzemeltetőjének nem kell a rendszer számára dedikált, saját erőforrásokat fenntartania. Így az ilyen rendszerek létrehozása és üzemeltetése egyszerűbb. A Grid eredeti víziójából itt az egyszerű kapacitás felajánlás valósult meg, mivel bárki csatlakoztathatja számítógépet egy egyszerű szoftver telepítésével és futtatásával, viszont a rendszert használni csak az üzemeltetői tudják: kevés, esetenként egyetlen felhasználója van. A DG és VC paradigma bár hasonlóknak tűnik, azonban két különböző használati esetet és környezetet céloz meg: a DG-ek alapvetően intézményen belül működnek, míg a VC rendszerek publikusan, az interneten. Így bár a DG-ek nagyban hasonlítanak a VC-hoz, de a két rendszer mégis alapvetően eltérő célokat szolgál. A SG-ekkel ellentétben – amelyekre létezik formális definíció és modell [69, 58] – nincs egyértelmű definíció ami alapján meghatározható a VC és DG rendszerek relációja, illetve a létező rendszerek egyértelműen kategorizálhatóak. Az összehasonlításuk (egymással és a SG-ekkel is) csak informális alapon létezik [65, 62, 60]. Valamint VC modell, formális alapokon, csak konkrét implementációból származtatva készült [59].

A VC rendszereket általában egy-egy számításiigényes probléma megoldására hozzák létre olyan kutatók, akiknek nincs hozzáférésük, illetve nem akarnak vagy nem tudnak dedikált infrastruktúrát fenntartani. Így ezek a rendszerek külön-külön gyűjtenek erőforrásokat az interneten keresztül. A legnépszerűbb VC keretrendszert – a BOINC-ot [64] – 80 különálló projekt használja, melyeket összesen 2.5 millió donor 8.5 millió számítógéppel támogat [42]. A projektek között vannak olyanok, amelyek folyamatosan generálják a feladatokat, emiatt még több erőforrásra lenne szükségük, és vannak olyanok, amelyek csak időszakosan. A projektek között nincs feladatelosztás, a donorok csatlakozhatnak egyenként a különböző projektekhez, megosztva közöttük

az erőforrásukat. A DG rendszereket intézményeken, egyetemeken belül használják, és a már meglévő irodai, vagy labor számítógépek szolgáltatják az erőforrásokat. Az egyetemek hierarchikus belső struktúrával rendelkeznek: a laborok tanszékekhez tartoznak, a tanszékek karokhoz. A tanszékek rendelkeznek saját erőforrásokkal, ugyanakkor a karoknak is vannak sajátjaik. Az intézményi belső struktúrát követve felállítható a DG-ek hierarchiája. Így képesek lennének a feladatmegosztásra egymás között, ezáltal közelebb kerülnének a Grid víziójában megfogalmazott erőforrás-megosztáshoz (felajánláshoz és használatához) [63]. A disszertációm következő részét ennek a feladatmegosztás kialakításának szentelem, melyhez felhasználom a már megalkotott modelljeimet.

A Cloud Computing (CC) [54, 61, 55, 52] – melynek megjelenését az Amazon Elastic Computing Cloud (EC2) kereskedelmi szolgáltatásának 2006-os elindulásához kötik – célja többek között a Grid víziójában szereplő számítási és háttértár kapacitáshoz való egyszerű hozzáférés. Ehhez olyan létező technológiákra épít, melyek a virtualizáció, üzleti-folyamatok, szolgáltatás-orientáltság vagy akár a Grid területéről származnak. Ennek ellenére a Grid és CC kapcsolata nem egyértelmű, mivel a CC tekinthető a Grid egy magas szintű absztrakciójának [57] is. Mindenesetre a CC gerincét – ugyanúgy mint az SG-eknél – a megbízható, dedikált erőforrások biztosítják. Felmerülhet a kérdés, hogy a SG és DG/VC rendszerek kapcsolata mintájára, a CC paradigmák alapján, a nem dedikált vagy önkéntesen felajánlott erőforrások segítségével, kialakítható-e Önkéntes Számítási Felhő (Volunteer Cloud Computing - VCC). A disszertációm fennmaradó részében ezt a problémakört vizsgálom. Először a CC NIST-féle definíciója [52] alapján azonosítom azokat a kihívásokat és megkötéseket, melyekre választ kell kapni önkéntes erőforrások segítségével történő CC szolgáltatás létrehozásához. Ez alapján megalkotom a VCC egy lehetséges definícióját, és az azonosított kihívásokra adok válaszokat.

Az első azonosított kihívás a rendszerek és az önkéntes erőforrások különbözősége (heterogeneitása): *(i)* a dedikált és felügyelt erőforrásokkal szemben az önkéntesek más-más hardver- és szoftverkönyezettel rendelkeznek és *(ii)* a VC rendszerek (és a DG-ek is) mind más-más követelményeket támasztanak a végrehajtandó feladatokkal (*job*-okkal) szemben. Rész megoldások léteznek, melyek vagy valamilyen programozási modell segítségével próbálják a rendszerek közötti különbséget elfedni [56] vagy az erőforrások heterogeneitásának kiküszöbölését tűzik ki célul [51, 49, 46, 66, 53], de ezek együttes használata nem megoldott, valamint egy-egy adott VC rendszer sajátosságaihoz kötöttek. Bemutatok egy, alulról építkező, általános megoldást a problémakörre. A második azonosított kihívás a nem dedikált erőforrások volatilitása. Egy-egy erőforrás csak korlátozottan, rövidebb-hosszabb ideig áll a rendszer rendelkezésére, valamint előfordulhat, hogy egyes erőforrások végleg eltűnnek. Ezek miatt az erőforrásokra kiosztott feladatok nem garantáltak, hogy határidőre vagy egyáltalán lefutnak. Ez különösen a kötegelt feladatok (*batch*) futtatásánál jelent problémát, ahol egy-egy köteg lefutási idejét egyetlen feladat lassú végrehajtási ideje (*round-trip time*) késleltetheti. Egy köteghez tartozó feladatok végrehajtási idejét ábrázolva láthatóvá válik, hogy néhány feladat végrehajtási ideje jóval magasabb a köteghez tartozó többi feladat idejéhez képest. Erre a jelenségre „farok-hatás”-ként (*tail effect*) is hivatkoznak [50, 47]. Olyan algoritmusokat és környezeteket hozok létre, amelyekkel a feladatkötegek végrehajtási ideje csökkenthető jelentős többleterőforrás felhasználása nélkül.

2. Kutatási módszertan

A disszertációm célja a DG és VC rendszerek vizsgálata, ezen belül formális modellekkel történő összehasonlítása; a rendszerek közötti feladatok (jobok) átjárhatóságának megvalósítása; az önkéntes felhők témakörén belül az önkéntes erőforrások és a rendszerek heterogenitásának kiküszöbölése; illetve a kötegelt feladatok végrehajtásánál jelentkező, úgynevezett „farok-hatás” csökkentése. Az eredményeim egy részét formális modellek és definíciók formájában írtam le, más részét pedig algoritmusokként.

A DG/VC rendszerek működésének szemantikáját az Abstract State Machines (ASMs) [67, 70] módszerrel modelleztem. Ez egy matematikailag megalapozott keretrendszer magas szintű rendszer tervezéshez és elemzéshez. Az ASM módszerrel könnyen elfedhetőek a tervezés során a nem fontos részletek, így az adott modellt koncepcionális szinten lehet megfogni, az implementációs részletekkel és tulajdonságokkal nem kell foglalkozni. Kiindulási modellnek egy, a tradicionális grideket definiáló ASM modellt vettem [69, 58]. Erre építkezve alkottam meg a formális modelljeimet. A modellek kapcsolatát *input-output-* és *trace-ekvivalenciával* vizsgálom.

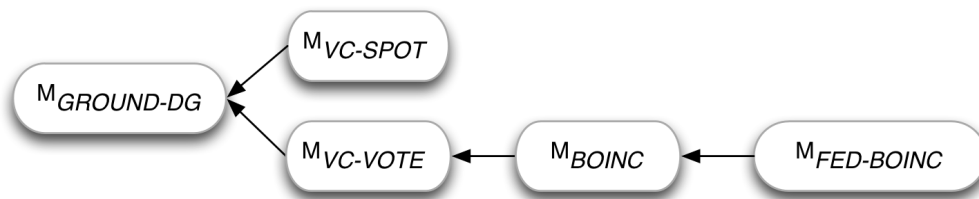
A feladatok (jobok) átjárhatósága és az önkéntes felhők témaköréhez tartozó eredményeim szintén modellek segítségével írtam le (nem ASM), ahol szükségesnek érzem kiegészítve az előző ASM alapú modelleket. A modelleket implementációk segítségével validáltam. Ezek C/C++ nyelven készültek, mivel a vizsgált rendszerek mind támogatják ezt valamilyen formában (például a BOINC API könyvtárakat biztosít). A megfelelően megírt C/C++ kód több operációs rendszere lefordítható, és a VC környezet sajátossága miatt az implementációk többségének különböző operációs rendszereken kell tudnia futni, előre telepített könyvtárak vagy futtató környezetek nélkül (egyetlen kivétel a VirtualBox megléte).

A kötegelt feladatok végrehajtásának vizsgálata kapcsán az algoritmusokat szintén formális módon írtam le. A validálást szintén implementációval és mérésekkel végeztem. Ehhez az SQL/Python nyelveket használtam: a feladatok állapot-lekérdezését és a hozzárendelések kezelését a BOINC rendszer sajátossága miatt SQL-ben készítettem, míg a vezérlő logikát Pythonban. A méréseket az EDGeS@home [43] és SZTAKI Desktop Grid [44] VC rendszereken végeztem valós körülmények között.

A méréseket saját naplózó rutinok segítségével hajtottam végre. Ezek mindig az eredeti implementáció instrumentált változatai. A környezet sajátosságából adódóan a méréseket elég a másodperc tartományban végezni, így az instrumentálás okozta hatást elhanyagolhatónak véltem (erről empirikusan megbizonyosodtam). A mérések elemzéséhez és vizualizálásához az R [45] környezetet használtam.

3. Új kutatási eredmények

Az eredményeimet két téziscsoportban foglalom össze. Az első téziscsoportban a DG/VC rendszereket hasonlítom össze formális modellek segítségével. Az itt bemutatott munkám célja, hogy a két rendszer-típusra (DG és VC) formális definíciót és referencia modellt adjak, melyek segítségével a meglévő rendszerek kategorizálhatóvá váljanak, illetve létező rendszerek formális modelljének elkészítéséhez nyújtsanak alapot. Legelőször egy informális szempontrendszer segítségével bemutatom a DG/VC rendszerek közötti különbséget, valamint megvizsgálom a létező implementációkat. Ezek után formális módszerekkel írom le a DG/VC rendszereket: az informális módszerekkel szemben, amelyek általában egy-egy karakterisztika alapján hasonlítanak össze, a formális módszerek segítségével a rendszerek szemantikáját ragadom meg. Ehhez a „hagyományos” grideket leíró, létező, modelltől indulok ki [58, 69], melyre M_{GRID} néven hivatkozom a disszertációban. A desktop grid rendszerek leírását egy formális modellen ($M_{GROUND-DG}$ a 3.1 ábrán), az önkéntes számítási rendszerek leírását két formális modellen ($M_{VC-VOTE}$ és $M_{VC-SPOT}$ a 3.1 ábrán) keresztül mutatom be. A modelleket összehasonlítom, ezáltal bemutatva a köztük levő viszonyt, illetve az eredeti grid modellel való kapcsolatukat. Egy negyedik modell (M_{BOINC} a 3.1 ábrán) segítségével, melyet az $M_{VC-VOTE}$ modelltől származtatok, egy létező rendszer – a BOINC – formális modelljét mutatom be. A modell megalkotása előtt felállítok egy követelmény-rendszert, mely tartalmazza a BOINC főbb jellemzőit. Megmutatom, hogy az elkészült modell megfelel ezen követelményeknek, illetve korrekt finomítása az $M_{VC-VOTE}$ modellnek. Az M_{BOINC} modell az azt megelőző modelljeim validálását (is) szolgálja. Ezen felül segítséget nyújthat más DG/VC rendszerek formális leírásához.



3.1. ábra. A DG/VC rendszerek kapcsán létrehozott formális modelljeim.

A második téziscsoportban a (i) DG/VC rendszerek közötti feladatmegosztás kialakításával, valamint a VCC kihívásai kapcsán az (ii) erőforrások és rendszerek heterogenitásának kiküszöbölésével és az (iii) önkéntes erőforrások volatilitása okozta „farok-hatás” csökkentésével foglalkozom. A feladatmegosztáshoz az M_{BOINC} modellt veszem alapul, mivel ez a modell tartalmazza a legtöbb részletet. Az elkészült modell ($M_{FED-BOINC}$ a 3.1 ábrán) tartalmaz egy biztonsági mechanizmust, mely a BOINC

mechanizmusát bővíti ki, illetve erre épül egy alkalmazás telepítési mechanizmus, mely a feladatok átjárhatóságának alapjául szolgál. A feladatmegosztás kapcsán három scenáriót azonosítottam (VC rendszerek közötti feladatmegosztás; hierarchikus DG rendszerek; és intézményi erőforrások által támogatott VC rendszerek), melyek közül a hierarchikus DG rendszerek segítségével részletezem a feladatmegosztás folyamatát.

Megvizsgáltam az NIST által alkotott definíciót a számítási felhőkre [52]. Ez alapján azonosítom azokat a kihívásokat és megköveteléseket, melyekre választ kell adni ahhoz, hogy önkéntes erőforrások segítségével CC szolgáltatást lehessen létrehozni. A disszertációmmal fennmaradó részében megalkotom a VCC egy definícióját, és az azonosított kihívásokra adok megoldásokat. Az első azonosított kihívás a rendszerek és erőforrások különböző szintű heterogenitása. Meghatározok különböző absztrakciós szinteket: a köztesréteg-absztrakció a rendszer és alkalmazás közötti definiált függőségeket teljesíti; a köztesrétegek-közötti-absztrakció a rendszerek közötti különbségeket fedi el; a környezet-absztrakció az önkéntes erőforrások eltérő szoftver környezete felett biztosít egységes környezetet, míg a teljes-absztrakció az eltérő szoftver környezet, különböző operációs rendszerek és részben hardver-környezet felett biztosít egy egységes környezetet. Bemutatom a DC-API, GenWrapper és GBAC környezeteket. A DC-API a köztesrétegek-közötti-absztrakcióra építve sikeresen elfedi a különböző vizsgált DG/VC rendszerek közötti különbségeket. Az általam alkotott GenWrapper környezet, mely a DC-API környezetre (így a köztesrétegek-közötti-absztrakcióra) valamint a köztesréteg- és környezet-absztrakcióra épít, és egy köztesréteg független környezetet biztosít, valamint elfedi a rendszer és alkalmazás közötti függőségeket. Az általam megtervezett GBAC környezet (a definiált köztesrétegek-közötti-, köztesréteg-, környezet- és teljes-absztrakció módszerekre építve) egy egységes szoftver és hardver környezetet biztosít feladatok futtatására önkéntes erőforrásokon, elfedi a különböző DG/VC rendszerek közötti különbségeket ezáltal, valamint teljesíti a rendszer és alkalmazás közötti követelményeket.

A második azonosított kihívás alapján, a VC rendszerekben az erőforrások volatilitása által a feladatkötegek végrehajtási idejének (*round-trip time*) elhúzódásának kiküszöbölésére, két környezetet és a kapcsolódó algoritmusokat mutatom be. Az első a VC rendszert fekete dobozként kezeli (nincs rálátása a belső működésére), azt kiegészítve, a futtatott feladatköteg aktuális állapotát figyelembe véve, a „problémásnak” azonosított feladatokból újabb példányokat hoz létre, ezzel segítve a sikeres végrehajtásukat. Megmutattam, hogy a módszerrel csökkenthető a feladatkötegek végrehajtási ideje, jelentős többletterhelés nélkül. A második környezet és a hozzá tartozó algoritmusok a BOINC belső ütemezését egészíti ki azzal, hogy egyes – adott feladatkötegekhez tartozó – feladatokat, megbízható erőforrásokra irányít át. Megmutattam, hogy a megfelelő feladatok átirányítása javítja a feladatköteg lefutási idejét.

A következőkben a téziseimet fogalmazom meg, illetve a hozzájuk kapcsolódó publikációimat sorolom fel.

I. Téziscsoport

Desktop Gridék és Önkéntes Számítási Rendszerek összehasonlítása és definíciója formális modelljeik alapján.

Készítettem egy szempontrendszert, mely a desktop gridék és az önkéntes számítási rendszerek főbb tulajdonságait írja le, lehetővé téve, hogy meglévő rendszereket egyértelműen rendszerezni lehessen. Több létező rendszert megvizsgáltam és a szempontrendszer alapján kategorizáltam, hogy önkéntes számítási rendszer vagy desktop grid. Ezt követően formális módszerekkel írom le a DG/VC rendszereket: az informális módszerekkel szemben, amelyek általában egy-egy karakterisztika alapján hasonlítanak össze, a formális módszerek segítségével a rendszerek szemantikáját ragadom meg. Ehhez a „hagyományos” grideket leíró, létező, modellből indulok ki [58, 69], melyre M_{GRID} néven hivatkozom a disszertációban.

I.1. Tézis: Desktop Gridék formális modellje és definíciója

Az M_{GRID} modell egy a Grid rendszerek leírására szolgáló, létező, formális modell. Az M_{GRID} modellre alapozva elkészítettem egy a desktop grid rendszerek leírására szolgáló formális modellt $M_{GROUND-DG}$ néven és igazoltam, hogy az általa leírt rendszerek megfelelnek az M_{GRID} modell által támasztott követelményeknek:

Tézis: *Az $M_{GROUND-DG}$ modell egy olyan kiterjesztése az M_{GRID} -nek, mely formalizált referencia modellt alkot a desktop grid rendszerek felépítéséről és alkalmazás-kezeléséről, ezáltal lehetővé teszi a desktop grid rendszerek elkülönítését az M_{GRID} által leírt hagyományos gridektől.*

A tézis értelmében az $M_{GROUND-DG}$ modell által leírt rendszerek is gridék az M_{GRID} modell értelmezésében.

Kapcsolódó publikációk: [8], [22], [36] és [37].

I.2. Tézis: Önkéntes Számítási Rendszerek formális modelljei és definíciója

Az $M_{VC-VOTE}$ és $M_{VC-SPOT}$ modellek egy-egy, az önkéntes számítási rendszerek leírására szolgáló általam megalkotott formális modell: Az $M_{VC-VOTE}$ a szavazásos eredményellenőrző módszert ("voting") leíró rendszerek egy modellje, míg az $M_{VC-SPOT}$ a spotellenőrzést ("spot-checking") használó rendszereké. A modellek lehetővé teszik tetszőleges számítási infrastruktúrák vizsgálatát, hogy desktop gridéknek és/ vagy önkéntes számítási rendszereknek minősülnek:

Tézis: *Az $M_{GROUND-DG}$ modellből származtatott és az önkéntes számítási rendszerek irányába szűkített $M_{VC-VOTE}$ és $M_{VC-SPOT}$ variánsok egy-egy olyan*

formális modellt reprezentálnak, melyek lehetővé teszik tetszőleges számítási infrastruktúrák kiértékelését az önkéntes számítási rendszerekkel szemben.

Az $M_{VC-VOTE}$ megfelel a hagyományos gridek modellezésére alkotott M_{GRID} formális modell követelményeinek, mivel megfelel az $M_{GROUND-DG}$ által felállított desktop grid definíciónak. Így az $M_{VC-VOTE}$ modell által leírt rendszerek is gridek az M_{GRID} modell értelmezésében, illetve desktop gridek az $M_{GROUND-DG}$ definíciója alapján, illetve megfelel az I.1 tézisben meghatározott desktop grid definíciójának. Az $M_{VC-SPOT}$ modellről belátható, hogy megfelel ugyanezen feltételeknek.

Kapcsolódó publikációk: [8], [22], [36] és [37].

I.3. Tézis: A BOINC formális modellje, és a modell relációja az $M_{VC-VOTE}$ modellel

Felállítottam egy követelményrendszert mely a BOINC legfontosabb jellemzőit tartalmazza. Megmutattam, hogy az általam alkotott M_{BOINC} modell, mely a BOINC önkéntes számítási rendszer egy lehetséges formális modellje, megfelel ezen jellemzőknek. Beláttam, hogy az M_{BOINC} modell az $M_{VC-VOTE}$ modell korrekt finomítása, így önkéntes számítási rendszereknek, desktop gridnek és gridnek minősül. Valamint megmutattam, hogy az M_{BOINC} modell az I.2 tézisben meghatározott feltételeknek megfelel:

Tézis: *Az M_{BOINC} modell az $M_{VC-VOTE}$ modell korrekt finomítása, továbbá az M_{BOINC} modell a BOINC szoftver rendszer belső felépítését és működését hűen reprezentálja, ebből következően a BOINC szoftver rendszer önkéntes számítási rendszernek, desktop gridnek és gridnek is minősül.*

Kapcsolódó publikációk: [2], [8], [22], [36] és [37].

II. Téziscsoport

Desktop Gridek és Önkéntes Számítási Rendszerek közötti feladatmegosztás, valamint Önkéntes Számítási Felhők kialakítása.

A desktop gridek és önkéntes számítási rendszerek függetlenül és elszigetelten működnek.

Valamint jellemző rájuk, hogy szakaszosan tudnak a klienseknek munkát biztosítani (például egy-egy kísérlet csak időszakonként fut). Ilyenkor a hozzájuk kapcsolódó kliensek, amennyiben csatlakoztak más rendszerekhez, máshonnan kaphatnak munkát. A kliensek saját hatáskörükben döntenek, mely projektetől fogadnak el munkát, így akkor is kaphatnak mástól, ha az adott projektnek van kiadható feladata. Amennyiben nem

csatlakoztak (vagy nem tudnak csatlakozni) akkor a számítási kapacitásuk ez időre elveszik, mivel ez a kapacitás nem tárolható.

Az önkéntes számítási rendszerek heterogén konfigurációjú és kapacitású erőforrásokat gyűjtenek össze. Ezen felül minden rendszernek megvannak a sajátosságai, illetve követelményei a rajtuk futtatandó (tudományos) alkalmazások felépítésével és működésével kapcsolatban. A felhő paradigma ezzel ellentétben homogén erőforrásokat és környezeteket ígér: homogén erőforrás alatt a processzor architektúra, elérhető memória és diszket kell érteni. Egy infrastruktúra szolgáltatás felhőnél (Infrastructure as a Service - IaaS) a kliens a szükséges mennyiséget kérheti, és (a paradigma alapján) tetszőleges mennyiségű példányt kérhet. A homogén környezet pedig a szabadon választható operációs rendszert, könyvtárakat és egyéb komponenseket jelöli. Ezzel szemben az önkéntes számítási rendszereknél a felajánlott erőforrások gyakorlatilag bármilyen hardver konfiguráció és architektúra lehet (amit az adott rendszer támogat), illetve a szoftver környezet is hasonlóan széles spektrumú.

II.1. Tézis: Desktop Gridok és Önkéntes Számítási Rendszerek közötti feladatmegosztás

Az általam definiált $M_{FED-BOINC}$ modell részét képező automatikus alkalmazás telepítési módszer és biztonsági mechanizmus integrációjának segítségével kialakított megoldás megvalósítja a desktop gridok és önkéntes számítási rendszerek munkacsomagjainak automatikus átjárhatóságát különböző projektek között. Valamint kidolgoztam scenáriókat a módszer lehetséges hasznosítására. A kidolgozott mechanizmusokat a BOINC önkéntes számítási rendszer és egyik azonosított scenárió segítségével mutatom be:

Tézis: *Az általam definiált automatikus alkalmazás telepítési módszer lehetővé teszi munkacsomagok automatikus átjárhatóságát desktop grid és/vagy önkéntes számítási rendszerek között.*

Kapcsolódó publikációk: [4], [6], [9], [10], [11], [12], [14], [17], [33], [38], és [40].

II.2. Tézis: Önkéntes Felhő Rendszerek

Definiáltam az önkéntes számítási rendszereknél (i) a köztesrétegek-közötti-, (ii) a köztesréteg-, (iii) a környezet- és (iv) a teljes-absztrakciós szinteket. A köztesréteg-absztrakció a rendszer és alkalmazás közötti definiált függőségeket teljesíti. A köztesrétegek-közötti-absztrakció a rendszerek közötti különbségeket fedi el. A környezet-absztrakció az önkéntes erőforrások eltérő szoftver környezetét fedi el, míg a teljes-absztrakció az eltérő szoftver környezet, különböző operációs rendszerek és részben hardver-környezet felett biztosít egy egységes környezetet. Ezen absztrakciókra támaszkodva megterveztem a DC-API, GenWrapper és GBAC környezeteket.

Az általam megtervezett GBAC környezet – mely a definiált absztrakciós módszerekre épül (lsd. i-iv) – egy egységes szoftver és hardver környezetet biztosít feladatok futtatására

önkéntes erőforrásokon, elfedi a különböző desktop gridek és önkéntes számítási rendszerek közötti követelmény-különbségeket, az eltérő környezeteket, valamint teljesíti a rendszerek és az adott alkalmazás között fennálló rendszer-követelményeket:

Tézis: *A GBAC környezet egy egységes szoftver és hardver környezetet biztosít munkacsomagok futtatására önkéntes számítási rendszerek erőforrásain, továbbá megalapozza egy új számítási infrastruktúra, az önkéntes számítási felhő rendszerek, virtualizált erőforrás kezelését.*

Kapcsolódó publikációk: [5], [7], [12], [14] [16], [18], [19], [20], [21], [23], [24], [25], [27], [28], [29], [30], [31], [32], [34], [35], [39] és [41].

II.3. Tézis: Feladatköteg végrehajtási idő csökkentése Önkéntes Számítási Rendszereken

Kidolgoztam munkacsomag kezelő algoritmusokat, melyek csökkentik az önkéntes számítási rendszereken ismert farok-hatás jelenség feladatkötegek végrehajtási idejére gyakorolt hatását:

Tézis: *A különböző módszereken alapuló - általam kidolgozott - "Adapt redundancy", "Prioritize delayed jobs" és "Prioritize job sources" algoritmusok az önkéntes számítási rendszereken tompítják a farok-hatás jelenség feladatkötegek végrehajtási idejére gyakorolt negatív hatását a meghatározott körülményektől függően.*

Mérésekkel megmutattam, hogy az algoritmusok valós körülmények között – a SZTAKI Desktop Grid és EDGeS@home önkéntes számítási projekteken – sikeresen javítják a feladatkötegek végrehajtási idejét. A javítás és az algoritmusok hatékonysága függ bizonyos körülményektől, melyeket disszertációmban részletezek.

Kapcsolódó publikációk: [1], [3], [13], [15] és [26].

4. Gyakorlati alkalmazás

A DG/VC rendszerek jellemzésére alkotott formális modellek lehetővé teszik – a BOINC-hoz hasonló módon – más rendszerek modellezését, meglévő rendszerek kategorizálását.

Az automatikus alkalmazás telepítési és az ismertett biztonsági módszer C/C++ implementációja a BOINC VC rendszer részét képezi 2008 óta. A munkacsomagok automatikus átjárhatóságát megvalósító implementáció a HAGRID¹ K+F projekt keretében hasznosult: a projekt résztvevőinek különböző rendszereken alapuló DG és VC infrastruktúráját kötötte össze a megoldásom egy meteorológiai alkalmazás futtatását segítve [36].

A II.2 tézis kapcsán definiált DC-API/GenWrapper és GBAC környezetek közül a DC-API implementációja a hivatalos BOINC API-ját kiegészítve a BOINC rendszer részét képezi, a beolvasztás véglegesítése folyik jelenleg. A GBAC és GenWrapper nyílt forráskódú projektként érhetőek el²³. A GBAC a közeljövőben a BOINC VBoxWrapper⁴ hivatalos része lesz, a beolvasztása folyamatban van. A DC-API/ GenWrapper és GBAC segítségével tudomásom szerint eddig 58 tudományos alkalmazást adaptáltak BOINC és XtremWeb alapú DG/VC rendszerekre. Az alkalmazások listája és rövid ismertetője, az adaptálás módszerével együtt, megtalálható a disszertációm A.1. függelékében. Ezek közül kiemelném a lista végéről (55. számú) „*patterncount*” nevű GenWrapperrel adaptált alkalmazást, mely egy, a Bruneli Egyetemre benyújtott és elfogadott, Ph.D. disszertáció szerves részét képezi [48].

Az eredményeimet továbbá az EDGeS⁵, EDGI⁶, DEGISCO⁷, CancerGrid⁸, IDGF-SP⁹, BioVeL¹⁰ európai; a WEB2GRID¹¹ és az AGRATÉR¹² hazai K+F projektek hasznosították: az alkalmazások egy részét ezen projektek adaptálták, illetve a DC-API/GenWrapper és GBAC rendszerek fejlesztését is támogatták.

A II.3 tézis alapját képező algoritmusok implementációi K+F projektek keretein belül működtetett, és az MTA SZTAKI által felügyelt, BOINC alapú VC rendszereken – a SZTAKI Desktop Griden [44] és az EDGeS@home-on [43] – futnak.

¹http://www.sztaki.hu/tudomany/projektek/projekt_informaciok/?uid=00188, elérés dátuma: 2013-01-01

²<http://genwrapper.sourceforge.net>, elérés dátuma: 2014-05-26

³<http://gbac.sourceforge.net>, elérés dátuma: 2014-05-26

⁴<http://boinc.berkeley.edu/trac/wiki/VboxApps>, elérés dátuma: 2014-10-09

⁵<http://edges-grid.eu/>, elérés dátuma: 2013-01-01

⁶<http://edgi-project.eu/>, elérés dátuma: 2013-01-01

⁷<http://degisco.eu/>, elérés dátuma: 2013-01-01

⁸<http://cancergrid.eu/>, elérés dátuma: 2009-01-01

⁹<http://idgf-sp.eu/>, elérés dátuma: 2014-05-26

¹⁰<http://www.biovel.eu/>, elérés dátuma: 2016-02-07

¹¹<http://www.egroup.hu/main/en/research/web2grid>, elérés dátuma: 2014-05-26

¹²<http://agrater.hu/>, elérés dátuma: 2016-02-07

Publikációs lista és irodalomjegyzék

Saját publikációk - Folyóirat cikk

- [1] Jozsef Kovacs, Attila Csaba Marosi, Adam Visegradi, Zoltan Farkas, Peter Kacsuk, and Robert Lovas. “Boosting gLite with cloud augmented volunteer computing”. In: *Futur. Gener. Comput. Syst.* 43-44 (2015), pp. 12–23. ISSN: 0167739X. DOI: 10.1016/j.future.2014.10.005. URL: <http://www.sciencedirect.com/science/article/pii/S0167739X14001897>. **(Impact Factor: 2.786)**
- [2] Attila Csaba Marosi and Róbert Lovas. “Defining volunteer computing: a formal approach”. In: *Comput. Res. Model.* 7.3 (2015), pp. 565–571. ISSN: 2076-7633. URL: <http://eprints.sztaki.hu/8361/%20http://crm-en.ics.org.ru/journal/issue/167/>.
- [3] Máté Pataki and Attila Csaba Marosi. “Searching for Translated Plagiarism with the Help of Desktop Grids”. In: *J. Grid Comput.* 11.1 (2013), pp. 149–166. ISSN: 1570-7873. DOI: 10.1007/s10723-012-9224-5. URL: <http://dx.doi.org/10.1007/s10723-012-9224-5>. **(Impact Factor: 1.667)**
- [4] Miklós Kozlovsky, Krisztián Karoczkai, István Marton, Ákos Balasko, Attila Csaba Marosi, and Péter Kacsuk. “Enabling generic distributed computing infrastructure compatibility for workflow management systems”. In: *Comput. Sci.* 13.3 (2012), pp. 61–78.
- [5] Attila Marosi, József Kovács, and Peter Kacsuk. “Towards a volunteer cloud system”. In: *Futur. Gener. Comput. Syst.* (2012), pages. ISSN: 0167-739X. DOI: 10.1016/j.future.2012.03.013. URL: <http://www.sciencedirect.com/science/article/pii/S0167739X12000660>. **(Impact Factor: 1.864)**
- [6] P. Kacsuk, J. Kovacs, Z. Farkas, Attila Csaba Marosi, and Z. Balaton. “Towards a Powerful European DCI Based on Desktop Grids”. In: *Journal of Grid Computing* 9 (2 2011). 10.1007/s10723-011-9186-z, pp. 219–239. ISSN: 1570-7873. URL: <http://dx.doi.org/10.1007/s10723-011-9186-z>. **(Impact Factor: 1.31)**
- [7] Attila Csaba Marosi, Jozsef Kovacs, and Peter Kacsuk. “Utilizing the power of Desktop Grid systems by Web 2.0 communities”. In: *Infocommunications Journal* 3.2 (2011), pp. 1–8. URL: <http://eprints.sztaki.hu/6731/>.
- [8] Peter Kacsuk, Jozsef Kovacs, Zoltan Farkas, Attila Csaba Marosi, Gabor Gombas, and Zoltan Balaton. “SZTAKI Desktop Grid (SZDG): A Flexible and Scalable Desktop Grid System”. English. In: *Journal of Grid Computing* 7 (4 2009), pp. 439–461. ISSN: 1570-7873. DOI: 10.1007/s10723-009-9139-y. URL: <http://dx.doi.org/10.1007/s10723-009-9139-y>.

- [9] Etienne Urbah, Peter Kacsuk, Zoltan Farkas, Gilles Fedak, Gabor Kecskemeti, Oleg Lodygensky, Attila Marosi, Zoltan Balaton, Gabriel Caillat, Gabor Gombas, et al. “Edges: Bridging EGEE to BOINC and XtremWeb”. In: *J. Grid Comput.* 7.3 (2009), pp. 335–354.
- [10] Zoltán Balaton, Zoltán Farkas, Gábor Gombás, Péter Kacsuk, Róbert Lovas, Attila Csaba Marosi, Ad Emmen, Gábor Terstyánszky, Tamás Kiss, Ian Kelley, Ian Taylor, Oleg Lodygensky, Miguel Cardenas-Montes, Gilles Fedak, and Filipe Araujo. “EDGeS: The Common Boundary Between Service and Desktop Grids”. In: *Parallel Process. Lett.* 18.3 (Sept. 2008), pp. 433–445.
- [11] Gábor Gombás, Attila Csaba Marosi, and Zoltán Balaton. “Grid Application Monitoring and Debugging Using the Mercury Monitoring System”. In: *Lecture Notes in Computer Science* 3470 (2005). Ed. by Peter M.A. Sloot, Alfons G. Hoekstra, Thierry Priol, Alexander Reinefeld, and Marian Bubak, pp. 193–199. DOI: [10.1007/11508380_21](https://doi.org/10.1007/11508380_21). URL: http://dx.doi.org/10.1007/11508380%7B%5C_%7D21. (Impact Factor: 0.302)

Saját publikációk - Könyvfejezet

- [12] Attila Csaba Marosi and Peter Kacsuk. “Volunteer Clouds: From Volunteer Computing to interconnected infrastructures”. In: *Dev. Interoper. Fed. Cloud Archit.* Ed. by Gabor Kecskemeti, Attila Kertesz, and Zsolt Nemeth. IGI Global, 2016, (ACCEPTED). ISBN: 9781522501534. DOI: [10.4018/978-1-5225-0153-4](https://doi.org/10.4018/978-1-5225-0153-4). URL: <http://www.igi-global.com/book/developing-interoperable-federated-cloud-architecture/142192>.
- [13] Péter Kacsuk, Zoltán Farkas, József Kovács, Ádám Visegrádi, Attila Csaba Marosi, Róbert Lovas, Gabor Kecskemeti, Zsolt Nemeth, and Mark Gergely. “Desktop Grid in the Era of Cloud Computing”. In: *Grid Comput. Tech. Futur. Prospect.* Ed. by Jorge G. Barbosa and Inês Dutra. Nova Science Publishers, 2015, pp. 187–206. ISBN: 978-1-63482-326-5.
- [14] Peter Kacsuk, Attila Marosi, Lovas Robert, and Jozsef Kovacs. “Supporting Web 2.0 Communities by Volunteer Desktop Grids”. In: *Deskt. Grid Comput.* Ed. by Gilles Fedak. Chapman and Hall/CRC, June 2012, pp. 287–307. ISBN: 978-1-4398-6214-8. DOI: [10.1201/b12206-16](https://doi.org/10.1201/b12206-16). URL: <http://www.crcnetbase.com/doi/abs/10.1201/b12206-16>.
- [15] P. Kacsuk, Marosi Attila Csaba, M. Kozlovszky, S. Acs, and Z. Farkas. “Parameter Sweep Job Submission to Clouds”. In: *Grids, Clouds and Virtualization.* Ed. by Massimo Cafaro and Giovanni Aloisio. Computer Communications and Networks. [10.1007/978-0-85729-049-6_6](https://doi.org/10.1007/978-0-85729-049-6_6). Springer London, 2011, pp. 123–141. ISBN: 978-0-85729-049-6. URL: http://dx.doi.org/10.1007/978-0-85729-049-6_6.

- [16] G. Kecskemeti, A. Kertesz, Attila Csaba Marosi, and P. Kacsuk. “Interoperable Resource Management for establishing Federated Clouds”. In: *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice*. Ed. by M. Villari, I. Brandic, and F. Tusa. IGI Global (USA), 2011, pp. 18–35. DOI: [doi:10.4018/978-1-4666-1631-8.ch002](https://doi.org/10.4018/978-1-4666-1631-8.ch002).
- [17] Zoltan Farkas, Attila Csaba Marosi, and Peter Kacsuk. “Job Scheduling in Hierarchical Desktop Grids”. English. In: *Remote Instrumentation and Virtual Laboratories*. Ed. by Franco Davoli, Norbert Meyer, Roberto Pugliese, and Sandro Zappatore. Springer US, 2010, pp. 79–97. ISBN: 978-1-4419-5595-1. DOI: [10.1007/978-1-4419-5597-5_8](https://doi.org/10.1007/978-1-4419-5597-5_8). URL: http://dx.doi.org/10.1007/978-1-4419-5597-5_8.
- [18] Attila Csaba Marosi, Zoltan Balaton, Peter Kacsuk, and Daniel Drotos. “SZTAKI Desktop Grid: Adapting Clusters for Desktop Grids”. In: *Remote Instrumentation and Virtual Laboratories*. Ed. by Franco Davoli, Roberto Pugliese, Norbert Meyer, and Sandro Zappatore. 10.1007/978-1-4419-5597-5_12. Springer US, 2010, pp. 133–144. ISBN: 978-1-4419-5597-5. URL: http://dx.doi.org/10.1007/978-1-4419-5597-5_12.
- [19] Attila Csaba Marosi, Gabor Gombas, Zoltan Balaton, and Peter Kacsuk. “Enabling Java applications for BOINC with DC-API”. In: *Distributed and Parallel Systems*. Ed. by Péter Kacsuk, Róbert Lovas, and Zsolt Németh. 10.1007/978-0-387-79448-8_1. Springer US, 2008, pp. 3–12. ISBN: 978-0-387-79448-8. URL: http://dx.doi.org/10.1007/978-0-387-79448-8_1.

Saját publikációk - Konferencia közlemény

- [20] József Kovács, Géza Ódor, Attila Marosi, and Ádám Kornafeld. “Exploring University Classes in Nonequilibrium Systems on SZTAKI Desktop Grid”. In: *Third Almer. Deskt. Exp. Work*. Almere: AlmereGrid, pp. 13–16.
- [21] Ferenc Horváth, Péter Ittzés, Dóra Ittzés, Zoltán Barcza, Laura Dobor, Dóra Hidy, and Attila Csaba Marosi. “Supporting environmental modelling with Taverna workflows, web services and desktop grid technology”. In: *Proc. 7th Int. Congr. Environ. Model. Softw.* Ed. by D.P. Ames, N.W.T. Quinn, and A.E. Rizzoli. San Diego, 2014, pp. 1–7. URL: http://www.iemss.org/sites/iemss2014/papers/iemss2014%7B%5C_%7Dsubmission%7B%5C_%7D220.pdf.
- [22] Attila Csaba Marosi and Zsolt Nemeth. “Two Sides of a Coin: Formalizing Volunteer and Desktop Grid Computing”. In: *Proc. Cracow Grid Work. 2013*. Ed. by M. Bubak, M. Turala, and K. Wiatr. Kraków: ACK CYFRONET AGH, 2013, pp. 69–70. ISBN: 978-83-61433-08-8.
- [23] A. Kertesz, G. Kecskemeti, M. Oriol, A. Marosi, X. Franch, and J. Marco. “Integrated Monitoring Approach for Seamless Service Provisioning in Federated Clouds”. In: *In proc. of the 20th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP'12)*. IEEE CS, 2012, pp. 567–574.

- [24] Sandor Acs, Miklos Kozlovszky, Attila Csaba Marosi, and Zoltan Balaton. “Amazon EC2 infrastruktúrán futó dinamikusan skálázható PBS klaszter”. In: *Networkshop 2011. Kaposvár, 2011*. Kaposvár: NIIF, Apr. 2011, pp. 1–11. URL: <http://eprints.sztaki.hu/6629/>.
- [25] Attila Csaba Marosi. “Web2Grid: Desktop Grid a Web 2.0 szolgálatában”. In: *Networkshop 2011. Kaposvár, 2011*. 2011, pp. 1–7. URL: <http://eprints.sztaki.hu/6719/>.
- [26] Attila Csaba Marosi and P. Kacsuk. “Workers in the Clouds”. In: *Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on*. Feb. 2011, pp. 519–526. DOI: [10.1109/PDP.2011.79](https://doi.org/10.1109/PDP.2011.79).
- [27] Attila Csaba Marosi, Gabor Kecskemeti, Attila Kertesz, and Peter Kacsuk. “FCM: an Architecture for Integrating IaaS Cloud Systems”. In: *In proc. of the Second International Conference on Cloud Computing, GRIDs, and Virtualization (Cloud Computing 2011)*. 2011, pages.
- [28] Jozsef Kovacs, Geza Odor, Attila Csaba Marosi, and Adam Kornafeld. “Exploring University Classes in Nonequilibrium Systems on SZTAKI Desktop Grid”. In: *Enter the Grid: Proceeding of the Third AlmereGrid Desktop Experience workshop*. AlmereGrid.nl, 2010, pp. 13–17.
- [29] Attila Csaba Marosi, Akos Balasko, and Livia Kacsukne Bruckner. “Porting the SEE-GRID EMMIL application into EDGeS infrastructure”. In: *Enter the Grid: Proceeding of the Third AlmereGrid Desktop Experience workshop*. AlmereGrid.nl, 2010, pp. 33–37.
- [30] Attila Csaba Marosi, Peter Kacsuk, Gilles Fedak, and Oleg Lodygensky. “Sandboxing for Desktop Grids Using Virtualization”. In: *Proceedings of the 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*. PDP '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 559–566. ISBN: 978-0-7695-3939-3. DOI: <http://dx.doi.org/10.1109/PDP.2010.90>. URL: <http://dx.doi.org/10.1109/PDP.2010.90>.
- [31] Attila Csaba Marosi, Attila Kovács, Péter Burcsi, and Ádám Kornafeld. “The BinSys application on the EDGeS infrastructure”. In: *Enter the Grid: Proceeding of the Third AlmereGrid Desktop Experience workshop*. AlmereGrid.nl, 2010, pp. 38–43.
- [32] Alejandro Rivero, Andres Bustos, Attila Csaba Marosi, Darrio Ferrer, and F. Serrano. “ISDEP, a fusion application deployed in the EDGeS project”. In: *Enter the Grid: Proceeding of the Third AlmereGrid Desktop Experience workshop*. AlmereGrid.nl, 2010, pp. 18–22.
- [33] Filipe Araujo, David Santiago, Diogo Ferreira, Jorge Farinha, Patricio Domingues, L Moura Silva, Etienne Urbah, Oleg Lodygensky, Haiwu He, Attila Csaba Marosi, et al. “Monitoring the EDGeS project infrastructure”. In: *Parallel Distrib. Process. 2009. IPDPS 2009. IEEE Int. Symp.* IEEE. 2009, pp. 1–8.

- [34] Attila Csaba Marosi, Zoltan Balaton, and Peter Kacsuk. “GenWrapper: A generic wrapper for running legacy applications on desktop grids”. In: *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. May 2009, pp. 1–6. DOI: [10.1109/IPDPS.2009.5161136](https://doi.org/10.1109/IPDPS.2009.5161136).
- [35] G Fedak, Haiwu He, O Lodygensky, Z Balaton, Z Farkas, G Gombas, P Kacsuk, R Lovas, A C Marosi, I Kelley, I Taylor, G Terstyanszky, T Kiss, M Cardenas-Montes, A Emmen, and F Araujo. “EDGeS: A Bridge between Desktop Grids and Service Grids”. In: *ChinaGrid Annu. Conf. 2008. ChinaGrid '08. Third*. Aug. 2008, pp. 3–9. DOI: [10.1109/ChinaGrid.2008.44](https://doi.org/10.1109/ChinaGrid.2008.44).
- [36] Zoltan Balaton, Gabor Gombas, Peter Kacsuk, Adam Kornafeld, Attila Csaba Marosi, Gabor Vida, Norbert Podhorszki, and Tamas Kiss. “SZTAKI Desktop Grid: a Modular and Scalable Way of Building Large Computing Grids”. In: *Workshop on Large-Scale and Volatile Desktop Grids, PCGrid 2007*. 2007, pp. 1–8.
- [37] Csaba Attila Marosi, Gabor Gombas, Zoltan Balaton, Peter Kacsuk, and Tamas Kiss. “SZTAKI Desktop Grid: Building a Scalable, Secure Platform for Desktop Grid Computing”. In: *CoreGRID Workshop - Making Grids Work*. Ed. by Marco Danelutto, Paraskevi Fragopoulou, and Vladimir Getov. Springer, 2007, pp. 365–376. ISBN: 978-0-387-78447-2.
- [38] Peter Kacsuk, Attila Csaba Marosi, Jozsef Kovacs, Zoltan Balaton, Gabor Gombas, Gabor Vida, and Adam Kornafeld. “SZTAKI Desktop Grid: a Hierarchical Desktop Grid System”. In: *Proceedings of the Cracow '06 Grid Workshop*. 2006.
- [39] Adam Kornafeld, Kovacs Attila, Burcsi Peter, Norbert Podhorszki, Attila Csaba Marosi, Gabor Vida, and Gabor Gombas. “Szuperszámítógépes Teljesítmény Szuperszámítógép Nélkül - A Binsys Projekt”. In: *Networkshop 2006*. Miskolc: NIIFI, 2006.
- [40] A. Cs. Marosi, G. Gombas, and Z. Balaton. “Secure Application Deployment in the Hierarchical Local Desktop Grid”. In: *Proc. of DAPSYS 2006 6th Austrian-Hungarian Workshop on Distributed and Parallel Systems*. Innsbruck, Austria, Sept. 2006, pp. 145–153.

Saját publikációk - Egyéb

- [41] Attila Csaba Marosi, Peter Kacsuk, Gilles Fedak, and Oleg Lodygensky. *Using Virtual Machines in Desktop Grid Clients for Application Sandboxing*. Tech. rep. TR-0140. Institute on Architectural Issues: Scalability, Dependability, Adaptability, CoreGRID - Network of Excellence, Aug. 2008. URL: <http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0140.pdf>.

Saját publikációk - Összegzés

	Közlemények száma	H-Index	Független idézések
MTMT ¹³	39	6	101
Scopus ¹⁴	25	7	148
Web of Science ¹⁵	12	3	34
Google Scholar ¹⁶	40	13	452 (Teljes)

Irodalomjegyzék

- [42] *BOINCstats/BAM!*. <http://boincstats.com>. URL: <http://boincstats.com> (visited on 01/01/2014).
- [43] *EDGeS@home*. <http://home.edges-grid.eu/>. URL: <http://home.edges-grid.eu/> (visited on 01/01/2013).
- [44] *SZTAKI Desktop Grid*. <http://szdg.lpds.sztaki.hu/szdg>. URL: <http://szdg.lpds.sztaki.hu/szdg> (visited on 01/01/2014).
- [45] *The R Project for Statistical Computing*. <http://www.r-project.org/>. URL: <http://www.r-project.org/> (visited on 05/23/2013).
- [46] Mohammadfazel Anjomshoa and Mazleena Salleh. “Overview on Clouds@ home: Virtualization Mechanism for Volunteer Computing”. In: *Int. Conf. "Parallel Distrib. Comput. Syst. PDCS 2014*. Vol. 2014. 2014, pp. 11–19. URL: <http://hpc-ua.org/pdcs-14/files/proceedings/2.pdf>.
- [47] Stephen C Winter, Christopher J Reynolds, Tamas Kiss, Gabor Z Terstyanszky, Pamela Greenwell, Sharron McEldowney, Sandor Acs, and Peter Kacsuk. “Buttressing volatile desktop grids with cloud resources within a reconfigurable environment service for workflow orchestration”. en. In: *J. Cloud Comput. Adv. Syst. Appl.* 3.1 (Jan. 2014), p. 1. ISSN: 2192-113X. DOI: 10.1186/2192-113X-3-1. URL: <http://www.journalofcloudcomputing.com/content/3/1/1>.
- [48] Mohammadmersad Ghorbani. “Computational analysis of CpG site DNA methylation”. PhD thesis. Brunel University, 2013. URL: <http://dspace.brunel.ac.uk/bitstream/2438/8217/1/FulltextThesis.pdf>.
- [49] Gary a. McGilvary, Adam Barker, Ashley Lloyd, and Malcolm Atkinson. “V-BOINC: The Virtualization of BOINC”. In: *2013 13th IEEE/ACM Int. Symp. Clust. Cloud, Grid Comput.* (May 2013), pp. 285–293. DOI: 10.1109/CCGrid.2013.14. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6546104>.

¹³<https://vm.mtmt.hu/search/slist.php?lang=0&AuthorID=10017749>

¹⁴<http://www.scopus.com/authid/detail.url?authorId=55664653500>

¹⁵https://apps.webofknowledge.com/CitationReport.do?product=UA&search_mode=CitationReport&SID=Z1BiRedaaV0wnrbbDda&page=1&cr_pqid=3&viewType=summary

¹⁶<https://scholar.google.hu/citations?user=PG30FrUAAAAJ&hl=hu>

- [50] Simon Delamare, Gilles Fedak, Derrick Kondo, and Oleg Lodygensky. “SpeQuloS: A QoS Service for BoT Applications Using Best Effort Distributed Computing Infrastructures”. In: *Proc. 21st Int. Symp. High-Performance Parallel Distrib. Comput. - HPDC '12*. New York, New York, USA: ACM Press, 2012, p. 173. ISBN: 9781450308052. DOI: [10.1145/2287076.2287106](https://doi.org/10.1145/2287076.2287106). URL: <http://dl.acm.org/citation.cfm?doid=2287076.2287106>.
- [51] D Ferreira, F Araujo, and P Domingues. “libboincexec: A Generic Virtualization Approach for the BOINC Middleware”. In: *Parallel Distrib. Process. Work. Phd Forum (IPDPSW), 2011 IEEE Int. Symp.* May 2011, pp. 1903–1908. DOI: [10.1109/IPDPS.2011.349](https://doi.org/10.1109/IPDPS.2011.349).
- [52] Peter Mell and Timothy Grance. *The NIST Definition of Cloud Computing*. Tech. rep. 800-145. Gaithersburg, MD: National Institute of Standards and Technology (NIST), Sept. 2011. URL: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
- [53] Carlos Aguado Sanchez, Jakob Blomer, Predrag Buncic, Gang Chen, John Ellis, David Garcia Quintas, Artem Harutyunyan, Francois Grey, Daniel Lombrana Gonzalez, Miguel Marquina, Pere Mato, Jarno Rantala, Holger Schulz, Ben Segal, Archana Sharma, Peter Skands, David Weir, Jie Wu, Wenjing Wu, and Rohit Yadav. “Volunteer Clouds and Citizen Cyberscience for LHC Physics”. In: *J. Phys. Conf. Ser.* 331.6 (2011), p. 62022. URL: <http://stacks.iop.org/1742-6596/331/i=6/a=062022>.
- [54] 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”. In: *Futur. Gener. Comput. Syst.* 25.6 (June 2009), pp. 599–616. ISSN: 0167739X. DOI: [10.1016/j.future.2008.12.001](https://doi.org/10.1016/j.future.2008.12.001). URL: <http://linkinghub.elsevier.com/retrieve/pii/S0167739X08001957>.
- [55] A Fox, R Griffith, and A Joseph. *Above the clouds: A Berkeley view of cloud computing*. Tech. rep. Electrical Engineering and Computer Sciences University of California at Berkeley, 2009. URL: <http://www-inst.cs.berkeley.edu/%7B~%7Ddcs10/sp11/lec/20/2010Fa/2010-11-10-CS10-L20-AF-Cloud-Computing.pdf>.
- [56] Eric M Heien, Yusuke Takata, Kenichi Hagihara, and Adam Kornafeld. “PyMW - A Python module for desktop grid and volunteer computing”. In: *IPDPS '09 Proc. 2009 IEEE Int. Symp. Parallel&Distributed Process.* Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–7. ISBN: 978-1-4244-3751-1. DOI: <http://dx.doi.org/10.1109/IPDPS.2009.5161132>.
- [57] Shantenu Jha, Andre Merzky, and Geoffrey Fox. “Using clouds to provide grids with higher levels of abstraction and explicit support for usage modes”. In: *Concurr. Comput. Pract. Exp.* 21.8 (2009), pp. 1087–1108.
- [58] Attila Kertész and Zsolt Németh. “Formal Aspects of Grid Brokering”. In: *Electron. Proc. Theor. Comput. Sci.* 14 (Dec. 2009), pp. 18–31. ISSN: 2075-2180. DOI: [10.4204/EPTCS.14.2](https://doi.org/10.4204/EPTCS.14.2). URL: <http://arxiv.org/abs/0912.2549v1>.

- [59] Yu Wang, Haiwu He, and Zhijian Wang. “Towards a formal model of volunteer computing systems”. In: *2009 IEEE Int. Symp. Parallel Distrib. Process.* IEEE, May 2009, pp. 1–5. ISBN: 978-1-4244-3751-1. DOI: 10.1109/IPDPS.2009.5161137. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5161137>.
- [60] Sungjin Choi, Rajkumar Buyya, Hongsoo Kim, Eunjoung Byun, and M Baik. “A Taxonomy of Desktop Grids and its Mapping to State-of-the-Art Systems”. In: *ACM Comput. Surv.* V (2008), pp. 1–61. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.8391%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>.
- [61] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. “Cloud Computing and Grid Computing 360-Degree Compared”. In: *2008 Grid Comput. Environ. Work.* (Nov. 2008), pp. 1–10. DOI: 10.1109/GCE.2008.4738445. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4738445>.
- [62] SungJin Choi, HongSoo Kim, EunJoung Byun, MaengSoon Baik, SungSuk Kim, ChanYeol Park, and ChongSun Hwang. “Characterizing and Classifying Desktop Grid”. In: *Seventh IEEE Int. Symp. Clust. Comput. Grid (CCGrid '07)* (May 2007), pp. 743–748. DOI: 10.1109/CCGRID.2007.31. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4215446>.
- [63] Péter Kacsuk, Norbert Podhorszki, and Tamás Kiss. “Scalable Desktop Grid System”. In: *High Perform. Comput. Comput. Sci. - VECPAR 2006* (2007), pp. 27–38. DOI: 10.1007/978-3-540-71351-7_3.
- [64] D P Anderson, C Christensen, and B Allen. “Designing a Runtime System for Volunteer Computing”. In: *SC 2006 Conf. Proc. ACM/IEEE.* Nov. 2006, p. 33. DOI: 10.1109/SC.2006.24.
- [65] Sungjin Choi, Hongsoo Kim, Eunjung Byun, and Chongsun Hwang. “A Taxonomy of Desktop Grid Systems Focusing on Scheduling”. In: (2006).
- [66] Brad Calder, Andrew a. Chien, Ju Wang, and Don Yang. “The entropia virtual machine for desktop grids”. In: *Proc. 1st ACM/USENIX Int. Conf. Virtual Exec. Environ. - VEE '05* (2005), p. 186. DOI: 10.1145/1064979.1065005. URL: <http://portal.acm.org/citation.cfm?doid=1064979.1065005>.
- [67] E Borger and Robert F Stark. *Abstract State Machines: A Method for High-Level System Design and Analysis*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003. ISBN: 3540007024.
- [68] Ian Foster and Carl Kesselman. *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier, 2003.
- [69] Z Németh and V Sunderam. “Characterizing grids: Attributes, definitions, and formalisms”. In: *J. Grid Comput.* (2003), pp. 9–23. URL: <http://link.springer.com/article/10.1023/A:1024011025052>.

- [70] Yuri Gurevich. “Evolving algebras: An attempt to discover semantics”. In: *Curr. Trends Theor. Comput. Sci.* (1993), pp. 1–27. URL: <http://books.google.com/books?hl=en%7B%5C%7Dlr=%7B%5C%7Ddid=kVhZDTKYa8MC%7B%5C%7Ddoi=fnd%7B%5C%7Dpg=PA266%7B%5C%7Ddq=EVOLVING+ALGEBRAS+:+AN+ATTEMPT+TO+DISCOVER+SEMANTICS%7B%5C%7Ddots=icN09xFapG%7B%5C%7Dsig=GVMXAgYnyS6bvIF0weICHuXYAYM>.

Rövidítések jegyzéke

- API Application Programming Interface
CC Cloud Computing
DC-API Distributed Computing API
DG Desktop Grid
GBAC Generic BOINC Application Client
NIST National Institute of Standards and Technology
SG Service Grid
VC Volunteer Computing
VCC Volunteer Cloud Computing