

# Munkában a böngészők

MOLNÁR GÁBOR, SCHNELL HENRIK, SZARVAS ATTILA, SZEBERÉNYI IMRE

BME Irányítástechnika és Informatika Tanszék  
 {schnell.henrik, molnar.gabor, szarvas.attila, szebi}@iit.bme.hu

Kulcsszavak: elosztott számítási rendszerek, grid, GridBee, HTML5, BOINC

**A gridek napjainkban jelentős szerepet töltenek be a nagyszabású tudományos számítások kivitelezésében.**

**A BOINC elosztott számítási menedzmentrendszer bejárattott és kiforrott eszközt biztosít az önkéntes gridek működtetéséhez, hiszen hétköznapi felhasználók is a kutatások rendelkezésére bocsáthatják számítógépeik kapacitását. Az eddigieknél jelentősen nagyobb közönséget szeretnénk megszólítani azáltal, hogy lehetővé tesszük a számítások böngészőkben való futtatását.**

## 1. Bevezetés

A kutatási feladatokhoz szükséges számítási kapacitást többnyire szuperszámítógépek, illetve az ezeket helyettesítő vagy kiegészítő elosztott számítási rendszerek biztosítják. Az elosztott rendszerekben a különálló számítógépek és számítógépfürtök általában önállóan oldják meg a számukra kiosztott feladatokat, majd egy kommunikációs csatornán továbbítják az eredményeket a központi gépre.

Az elosztott megoldások előnye a rugalmasság, a skálázhatóság és a költséghatékonyság. Lehetővé teszik, hogy sok, egyenként kisebb teljesítményű, kommersz egység révén összességében hatalmas számítási teljesítményre tegyünk szert. Az elosztott számítások gyakran alkalmazott architektúrája a *grid*, amely különálló számítógépek erőforrásait szervezi egy egységes rendszerbe, lehetővé téve, hogy a résztvevők valamennyi állomás együttes teljesítményét igénybe vegyék feladataik végrehajtásához.

Már az 1990-es évek második felében létrejöttek az első *közösségi számítási projektek*, amelyek önkéntes alapon szerveződtek és a felhasználók különböző kutatási célok érdekében kínálták fel saját erőforrásaikat. Ma már kevésbé ismert, bár akkoriban igen nagy port kavart az 1997-ben indított *distributed.net* projekt [1], amivel 250 nap alatt sikerült egy 56 bites RSA kulcsot megtörni, és ezzel meggyőzően demonstrálta az önkéntesen összeadott erőforrások létjogosultságát.

A számos önkéntes projekt közül az egyik legrégebbi és egyben legismertebb a számtalan donort maga mögött tudó, és még jelenleg is igen aktív *SETI@home* [2] projekt, ami a Berkeley egyetemen kifejlesztett BOINC-keretrendszer [3-6] révén éri el a donorok erőforrásait. A keretrendszer két fő komponensből áll: a számítási-igényes alkalmazásokat a donor számítógépén futtató *BOINC-kliensprogramból* és a központi ütemezésért felelős *BOINC-szerverből*. A BOINC-kliensszoftvere minden jelentős operációs rendszerhez elérhető és közel 40 tudományos kutatáshoz csatlakozhatunk a segítségével.

A kutatásokat támogató donorok száma önmagában véve figyelemre méltó (egy adott időpontban nagyságrendileg 300 000 aktív felhasználó van jelen a rendszerben [18]), a Web2-es robbanást követő, főként közösségi oldalak körül csoportosuló tömegekhez képest azonban eltörpül. Ebben vélhetőleg szerepet játszik az, hogy a BOINC-klienst telepíteni kell a donor számítógépére. A telepítés során a kliens beépül az operációs rendszerbe, ütemezője pedig alapértelmezés szerint indul és a háttérben folyamatosan fut. A kliens által futtatott alkalmazások korlátozásokkal ugyan, de elérhetik a donor gépnek erőforrásait és adatait is, ezért a biztonságos futás érdekében különösen nagy gondot kell fordítani a számítási alkalmazások ellenőrzésére.

A *Web2Grid projekt* [7] célja, hogy a Web2 közösség számára elérhetővé tegye és kiaknázza a grid technológiát, mind a megrendelői oldalnak (akik számítási-igényes futtatást szeretnének végrehajtani), mind az erőforrásukat felajánlani szándékozóknak, azaz a donoroknak.

A következőkben bemutatjuk a W2G projekt keretében fejlesztett GridBee [8] programkönyvtárat és a ráépülő webes klienst, amely lehetővé teszi a felhasználóknak, hogy mindössze egy böngésző segítségével, biztonságos környezetben elosztott számítási projekteken vehessenek részt.

## 2. A GridBee keretrendszer célkitűzése

A Web2Grid projekt keretében létrehozott *GridBee keretrendszer* egy JavaScript függvénykönyvtár, amely közösségi számítások futtatását teszi lehetővé a modern böngészőkben. Kiemelt jelentősége miatt a keretrendszerrel párhuzamosan egy BOINC modult, valamint egy webes felhasználói felületet is fejlesztettünk, amelyek lehetővé teszik, hogy egy böngészőben futó webes alkalmazással a BOINC kliens minden fontos funkcióját el-  
 lássuk.

A keretrendszert az általánosság és a modularitás követelményét szem előtt tartva, a klienstől teljesen füg-

getlenül fejlesztettük, így a későbbiekben könnyen építhetünk rá tetszőleges honlapba beágyazható, önműködően futó alkalmazásokat. A központi ütemező szerverek feladata a számításokat elvégző alkalmazások, és az azokhoz tartozó adatok elküldése a donoroknak, de ezekhez a szerverekhez érkeznek be a donorok által kiszámított eredmények is. A moduláris felépítésnek köszönhetően kisebb projektek esetén a központi ütemező szerepét a BOINC-nál egyszerűbb, saját fejlesztésű szerver is betöltheti egy megfelelő GridBee-modul segítségével. A tesztelés során ugyanakkor mi is a BOINC szerverét használtuk, így az ismertetett példákban is egy BOINC-szerver fogja ellátni ezt a feladatot.

Napjainkban jellemző az alkalmazások webes változatainak megjelenése, amelyek gyorsan nagy népszerűségekre tesznek szert. Ezeket a programokat nem kell telepíteni és operációs rendszertől függetlenül futtathatók egy támogatott böngészőben. A felhasználó adatai sem kötődnek fizikailag egy adott helyszínhez, fiókjába bárholnan bejelentkezhet.

Reményeink szerint a bemutatott megoldással sikerül minden eddiginél közelebb vinni a felhasználókhoz az elosztott számítások világát. A minél gördülékenyebb működés és a gyors elterjedés érdekében biztosítani kell, hogy a könyvtár működéséhez ne kelljen semmiféle kiegészítőt telepíteni, egy modern böngésző önmagában is képes legyen a futtatására. A szerverekkel való kommunikáción túl a könyvtár azt is lehetővé teszi, hogy a tudományos számításokat is a böngésző motorja végezze el anélkül, hogy a felhasználói élményt rontaná.

A folyamat egészének böngészőben tartásával két szempontból is elősegítjük a közösségi számítások elterjedését. Egyrészt a webes alkalmazást nem kell telepíteni, ezért a kíváncsi felhasználók gyorsan kipróbálhatják és további kööttségeket sem kell vállalniuk. Másrészt a böngésző által értelmezett nyelveken írt programok az operációs rendszertől elszigetelten futnak. A jelenlegi gyakorlattal ellentétben, ahol az operációs rend-

szer által futtatott, natív kódot alkalmaznak, ez komoly biztonsági előrelépés, amely segíthet megnyerni a felhasználók bizalmát, és csökkenthetjük a futtatni kívánt kódok ellenőrzésére fordított erőfeszítéseket is, ami a számítási feladatok megalkotóira hathat bátorítóan. A számítások operációs rendszertől való elszigetelésének igénye már korábban is felmerült, melynek egyik megoldása a kliens virtuális gépbe zárása volt [9].

Az általunk készített általános függvénykönyvtárat számos különböző módon felhasználhatjuk. A már említett, böngészőben futó BOINC-alkalmazás a legkézenfekvőbb megoldás, amelynek elkészítése így közvetlenül a Grid Bee projekt részét képezi. Az elképzelés a tudományos számítások iránt érdeklődő, részben jelenleg is aktív felhasználókat célozza meg. Egy már aktív felhasználó új, natív klienssel nem rendelkező számítógépeket vonhat be a számításokba, illetve meggyőzheti a telepítési procedúrától ódzkodó ismerőseit a csatlakozásról.

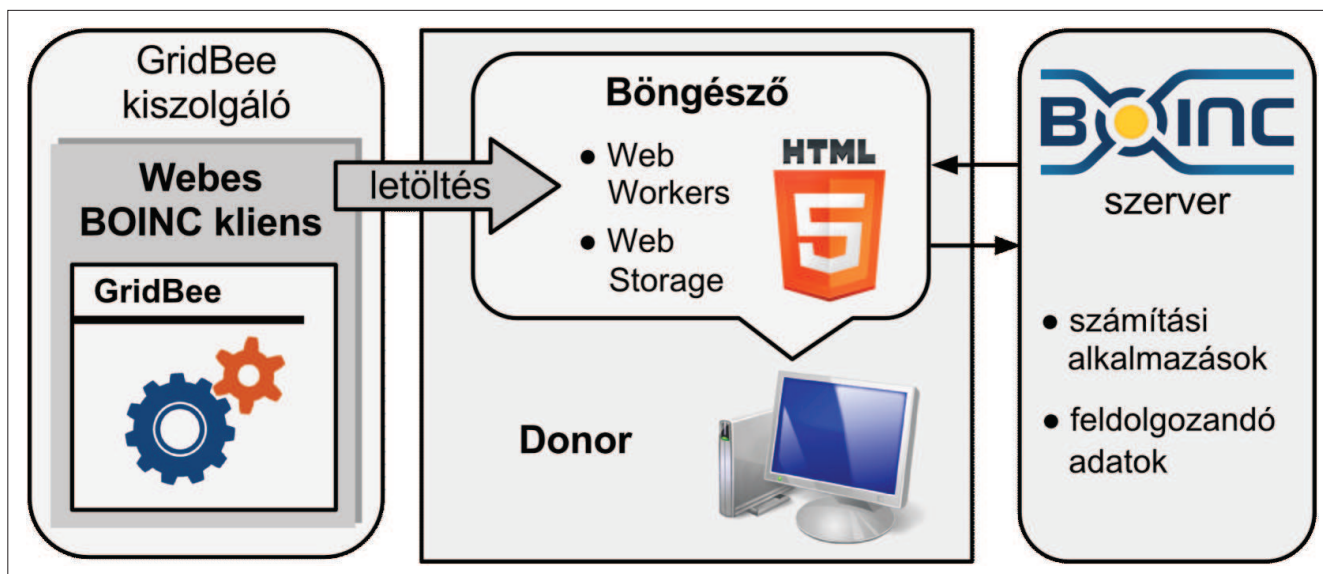
Egy másik elképzelés a könyvtár közösségi oldalakon futó alkalmazásokba való integrálása. Ha egy projekt tudományos célkitűzése képes felkelteni néhány ember figyelmét, akkor a kapcsolati hálón szétterjedve rövid idő alatt nagy felhasználói bázisra tehet szert.

További lehetőség az új generációs webes szolgáltatások és alkalmazások támogatása biztonságos, üzleti alapú grid platformmal. A könyvtár lehetővé teszi, hogy a számításokat tetszőleges profilú weboldalon vagy webes alkalmazás részeként a háttérben végezzük el. Különös gondot kell fordítani természetesen a felhasználók (donorok) tájékoztatására, és fel kell ajánlani számukra a lehetőséget, hogy a számítási feladatokat szüneteltesék, vagy teljes mértékben letiltsák.

### 3. Technológia

A GridBee minden olyan kliensoldali szolgáltatást megvalósít, amelyre a napjainkban alkalmazott, BOINC-ra épülő gridekkel való együttműködéshez szükség van.

1. ábra A GridBee szerepe a webes elosztott számítások rendszerében



A BOINC-modul felhasználásával a függvénykönyvtárra épülő alkalmazások ugyanazokhoz a BOINC-szerverekhez kapcsolódhatnak, amelyek már jelenleg is részt vesznek a különböző tudományos projektek kezelésében.

A projektek már ma is több különböző platformra lefordított kódot tartalmaznak annak érdekében, hogy a potenciális donorok minél nagyobb hányadát be tudják vonni a számításokba. A GridBee-kliensek kiszolgálásához a BOINC-szervereken be kell jegyezni a *javascript* platformot, illetve el kell készíteni a tudományos alkalmazást ezen a nyelven is. Egyszerűbb programok C-ről JavaScriptre való fordítását az Emscripten [10] segítségével automatikusan is el lehet végezni.

A webes BOINC-kliens letöltési címét meglátogatva az letöltődik a böngészőnkbe és a felhasználó parancsaitól függően, esetleg teljesen automatikusan felveszi a kapcsolatot a szerverrel. A számítási alkalmazás és a feldolgozandó adatok letöltését követően végrehajtja az előírt számításokat, az eredményt pedig feltölti a szervernek.

### 3.1. Nagyteljesítményű számítások JavaScriptben

A függvénykönyvtárat *haXe* [11] nyelven fejlesztjük, amely egy nyílt forrású, aktív közösséggel rendelkező, típusos, objektumorientált nyelv. Szolgáltatásai és a hozzá kapcsolódó fejlesztőeszközök megkönnyítik a nagyméretű projektek kezelését.

A könyvtárat JavaScriptre fordítjuk, amely így bármilyen kiegészítő nélkül képes futni a támogatott böngészőkben. A BOINC szerverekkel való kommunikációt, a komplex funkciókkal rendelkező alkalmazások kialakítását és a komoly számítások zökkenőmentes futtatását a böngészőkben most debütáló *HTML5* [12] szabványcsoport és a *Web Workers* szabvány teszi lehetővé.

A keretrendszer által a böngészőkben futtatható tudományos alkalmazások nyelve *JavaScript* vagy *ActionScript* is lehet. A projekt korai szakaszában megvizsgáltuk, hogy az egyes böngészők alkalmasak-e egyáltalán számításintenzív szkriptek futtatására. Az összeállított tesztek a lebegőpontos műveletvégzés, valamint a függvényhívások sebességére voltak érzékenyek. Az azonos tartalmú teszteket C++-ból fordított natív kóddal, illetve különböző böngészőkben végrehajtott Java Script parancsfájlokkal végeztük el (2. és 3. ábra).

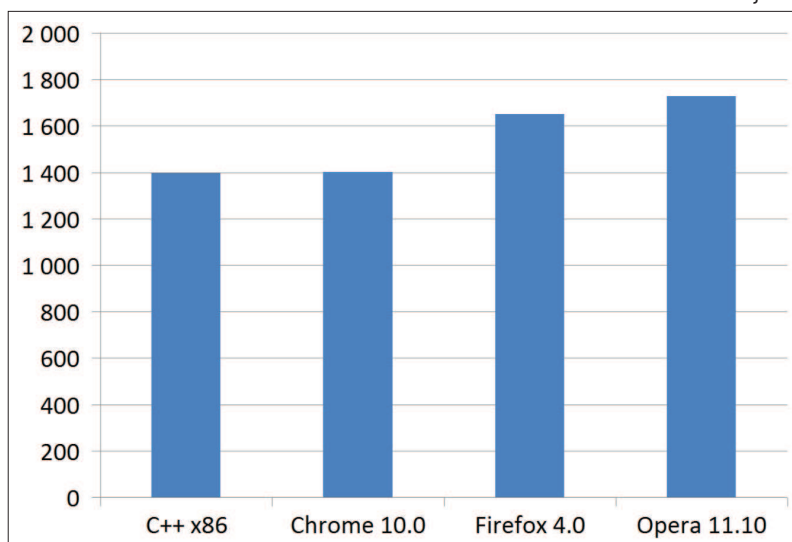
A mérések során biztató, sőt egészen meglepő eredményeket is tapasztaltunk. A legújabb böngészők *JavaScript*-végrehajtási sebessége az általunk vizsgált feladatok esetén a natív kódéval összemérhető. A böngészők a sebesség tekintetében egyértelműen alkalmassá váltak a komoly teljesítményt igénylő számítások elvégzésére is.

### 3.2. Többszálú kódvégrehajtás böngészőben

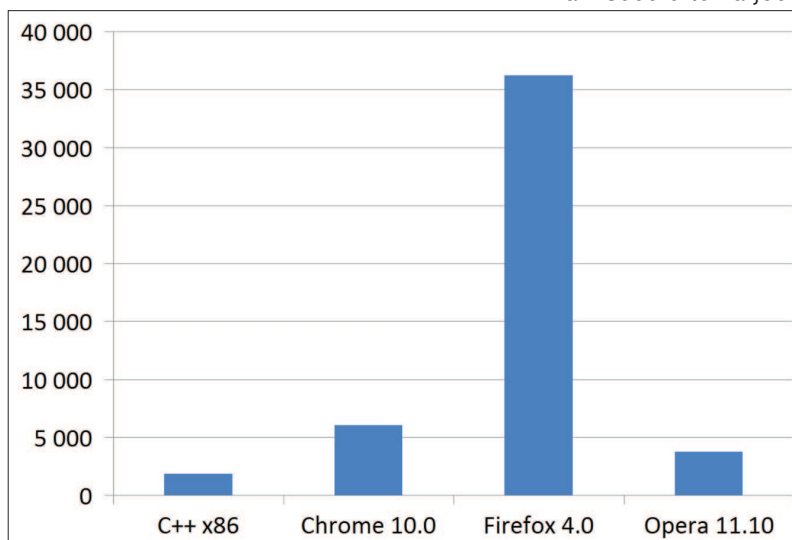
A *Web Workers* [13] technológiára részben a felhasználói élmény garantálása szempontjából van szükségünk. Egészen a közelmúltig a böngészők egy szálon hajtották végre kódjukat, és ugyanezen a szálon futtatták a parancsfájlokat is. Ez ahhoz vezetett, hogy az intenzív számításokat végző kódok „megfagyasztották” a honlap kezelőeszközeit, rosszabb esetben a böngésző felületét is. A *Web Workers* lehetővé teszi, hogy a keretrendszer ezeket a böngésző fő szálával párhuzamosan, attól elszigetelve futtassa. Ennek köszönhetően a számítások nem lassítják a felhasználói interfészt futtató szálat, ráadásul több *worker* indításával a többmagos processzorokat is teljes mértékben ki tudjuk használni.

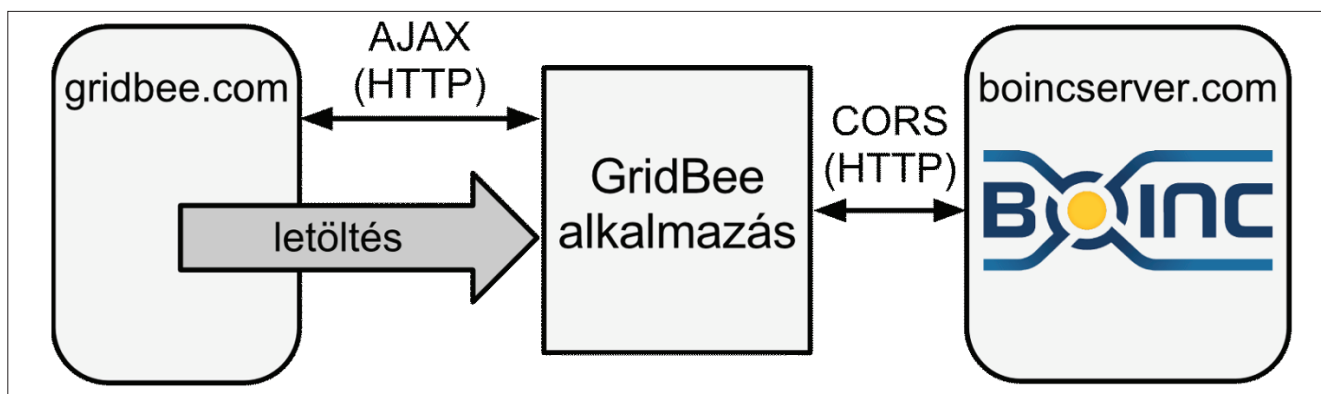
További előnyt jelent, hogy a *workerek* a honlaptól is elszigetelten futnak, a főablakbeli kóddal csak üzenetek formájában tarthatják a kapcsolatot. A számításokat így nemcsak az operációs rendszertől, hanem a keretrendszertől is el tudjuk választani. Az eddigiekhez

2. ábra  
A lebegőpontos teszt elvégzésének ideje (sec)  
– a kisebb érték a jobb



3. ábra  
A függvényhívási teszt elvégzésének ideje (sec)  
– a kisebb érték a jobb





4. ábra Kommunikáció a kliens, illetve a BOINC szerverrel

képest ez a biztonság terén is előrelépést jelent, amelynek köszönhetően nincs szükség a számítási alkalmazások kódjának tüzetes vizsgálatára.

### 3.3. Perzisztens adattárolás

A felhasználói beállítások és a számítások köztes eredményeinek tárolásához a HTML5 *Local Storage* [14] szabványa nyújt lehetőséget. A szabvány egy domainhez kötött perzisztens tárhelyet specifikál, melyben kulcs-érték párok formájában tárolhatunk adatokat. Mérete jelenleg böngészőtől függően 2-10 MB között mozog, amely a tipikusan számításintenzív feladatok adatainak tárolásához sokszor elegendő.

A Local Storage-ot megelőzően a kliens csak *cookie*-k formájában tárolhatott adatokat, amelyeket a böngésző minden HTTP lekérdezés során elküldött a szervernek. A Web Storage javascriptes utasítások segítségével kezelhető, a böngésző nem küldi el tartalmát a szervernek, így nem terheli vele a kommunikációt.

### 3.4. Kommunikáció a kiszolgáló szerverekkel

A GridBee keretrendszerre épülő webes alkalmazásokat annak készítői tetszőleges tárhelyen, tetszőleges domain alatt elhelyezhetik. A felhasználók például a *gridbee.com* domaint meglátogatva letöltik böngészőikbe a webes BOINC klienst, amely felveszi a kapcsolatot az ütemezésért felelős BOINC szerverrel. A továbbiakban az alkalmazásnak már csak a *boincserver.com*-mal szükséges kommunikálnia.

Alapesetben a JavaScript programok csak azzal a kiszolgálóval létesíthetnek kapcsolatot, amelyről a böngésző letöltötte őket. Ezt hívják *same origin policy*-nek [15], amely nyilvánvalóan nem korlátozza a *gridbee.com* állomás és a webes BOINC-kliens kapcsolatát. Ahhoz viszont, hogy az alkalmazás bármely más szerverrel is felvehesse a kapcsolatot, úgynevezett *cross-origin resource sharing (CORS)* [16] megoldást kell alkalmaznunk.

A CORS működéséhez a BOINC-szervert is futtató állomás webkiszolgálójában a tartalmak elérhetővé tételéhez el kell látni azokat egy engedélyező HTTP headerrel, amelyben megadjuk annak az állomásnak a címét, amely számára lehetővé kívánjuk tenni a CORS kapcsolatot. Az eljárás a BOINC-szerver számára teljesen átlátszó, annak fájljait módosítani nem kell.

## 3. Összefoglalás és kitekintés

Projektünk célja, hogy a böngészőket az elosztott számítási megoldások egy újabb platformjává téve az eddigieknél nagyobb kényelmet és rugalmasságot biztosítsunk a donorknak. Emellett azt is szeretnénk, ha a számítási feladatok tervezői és üzemeltetői kis módosításokkal ugyan, de továbbra is a már jól bevált eszközöket alkalmazhatnák.

Tapasztalataink alapján az elkészült GridBee keretrendszer az első elvárásnak sikeresen megfelelt, a BOINC modul pedig lehetővé teszi, hogy a tudományos elosztott számítások terén legnépszerűbb BOINC-számítási menedzsmentrendszerrel is együttműködjön. Méréseink szerint a JavaScript alkalmas lehet számításigényes tudományos alkalmazások futtatására, további vizsgálatokra van szükség ugyanakkor a már meglévő alkalmazások JavaScriptre való fordításának lehetőségéről. A továbbiakban azt is meg szeretnénk vizsgálni, milyen böngészőben futtatható alternatívái vannak a JavaScriptnek, amelyek együttműködhetnek a GridBee keretrendszerrel.

A Firefox 5.0 és a Chrome 10.0 már teljes mértékben alkalmas a GridBee függvénykönyvtárra épülő alkalmazások futtatására. A központi funkciókat sikeresen teszteltük az Internet Explorer 10 Platform Preview 2 esetében is. A szabványos megoldások alkalmazása révén idővel várhatóan minden fontos böngésző támogatni fogja a szükséges technológiákat [17].

A böngészők aktuális fejlettsége révén megoldásunk tökéletesen alkalmas az internetezők széles rétegének közösségi számításokba való bevonására.

A hagyományos, tudományos webalkalmazáson túl a közösségi oldalakba integrált megoldások és üzleti modellek is elképzelhetők. Ha sikerül minél több, a GridBee alkalmazó, böngészőben futtatható alkalmazást elindítani, akkor az önkéntes gridek rendelkezésére álló számítási teljesítmény nagyságrendileg növekedhet.

### Köszönetnyilvánítás

A munka a Nemzeti Technológia Program (TECH\_08-A2/2-2008-0097 WEB2GRID) támogatásával valósult meg.



## A szerzőkről



**MOLNÁR GÁBOR** a Budapesti Műszaki és Gazdaságtudományi Egyetem mérnök informatikus MSc szakos hallgatója „Hálózatok és szolgáltatások” szakirányon. A BME Közigazgatási Informatikai Központjában webalkalmazások fejlesztésével foglalkozik.



**SCHNELL HENRIK** a Budapesti Műszaki és Gazdaságtudományi Egyetem végzős hallgatója a BSc mérnök informatikus képzés Informatikai technológiák szakirányán. Kutatási területe a háromdimenziós számítógépes grafika, valamint a modern, HTML5 alapú webes technológiák. Szoftverterchnológia és objektumorientált szoftvertervezés témákban demonstrátori feladatokat lát el.



**SZARVAS ATTILA** 2011-ben a Budapesti Corvinus Egyetemen szerzett diplomát nemzetközi tanulmányok alapszakon. Jelenleg a Budapesti Műszaki és Gazdaságtudományi Egyetem villamosmérnöki alapszakának végzős hallgatója beágyazott- és ambiens rendszerek szakirányon. Szakdolgozatának témája az aktív zajcsökkentés, a BME Közigazgatási Informatikai Központjában webes fejlesztéssel foglalkozik.



**SZEBERÉNYI IMRE** a Budapesti Műszaki Egyetemen szerzett villamosmérnöki diplomát 1983-ban, majd PhD fokozatot 2003-ban. 1983-tól a BME Irányítástechnikai és Informatikai Tanszékének docense és a Közigazgatási Informatikai Központjának kutatás-fejlesztési igazgatóhelyettese. Kutatási területei közé tartoznak a párhuzamos programozás és párhuzamos algoritmusok alkalmazása, az elosztott számítási környezetek vizsgálata, valamint a grid és cloud rendszerek fejlesztése.

## Irodalom

- [1] Distributed.net projekt:  
[http://www.distributed.net/Main\\_Page/en](http://www.distributed.net/Main_Page/en)  
2011. április
- [2] SETI@HOME projekt:  
<http://setiathome.ssl.berkeley.edu/>
- [3] BOINC projekt:  
<http://boinc.berkeley.edu/>  
2011. április
- [4] BOINC scheduling server protocol:  
<http://boinc.berkeley.edu/trac/wiki/RpcProtocol/>  
2011. április
- [5] BOINC protocol overview:  
[http://www.boinc-wiki.info/Protocol\\_Overview/](http://www.boinc-wiki.info/Protocol_Overview/)  
2011. április
- [6] BOINC Web RPC:  
<http://boinc.berkeley.edu/trac/wiki/WebRpc>  
2011. április
- [7] Web2Grid projekt:  
<http://web2grid.econet.hu/>  
2011. április
- [8] GridBee kezdőoldal:  
<http://webcomputing.iit.bme.hu/>  
2011. április
- [9] Németh Dénes, Szeberényi Imre, Deák Szabolcs, Virtualizált Biztonságos BOINC, Networkshop 2009, Szeged, 2009. április 15-17.
- [10] Az Emscripten projekt honlapja:  
<https://github.com/kripken/emscripten>,  
2011. július
- [11] Haxe nyelv:  
<http://haxe.org/doc>  
2011. április
- [12] HTML5 technológiák támogatottsága böngészőkben:  
[http://www.caniuse.com/#cats=HTML5,JS\\_API](http://www.caniuse.com/#cats=HTML5,JS_API)  
2011. április
- [13] Web Workers:  
<http://www.w3.org/TR/workers/>  
2011. április
- [14] A Web Storage oldala a W3C-nál:  
<http://dev.w3.org/html5/webstorage/>  
2011. július
- [15] Same origin policy:  
[https://developer.mozilla.org/en/Same\\_origin\\_policy\\_for\\_JavaScript](https://developer.mozilla.org/en/Same_origin_policy_for_JavaScript)  
2011. április
- [16] Cross-Origin Resource Sharing:  
<http://www.w3.org/TR/cors/>  
2011. április
- [17] Usage share of web browsers:  
[http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp)  
2011. július
- [18] Aktív felhasználók száma a BOINC projektjeiben:  
<http://boinc.berkeley.edu/index.php>  
2011. augusztus