

# A KOPI Plágiumkereső terhelésének elosztása cloud környezetben

Micsik András, Pataki Máté, Garzó András  
MTA SZTAKI, 1111 Budapest, Lágymányosi utca 11.  
{micsik.andras, pataki.mate, garzo.andras}@sztaki.mta.hu

**Kivonat:** Az MTA SZTAKI Elosztott rendszerek Osztálya által fejlesztett KOPI Online Plágiumkereső és Információs Portál egy egyedülálló, nyílt szolgáltatás az internetező közönség számára, amely lehetővé teszi, hogy a felhasználók saját dokumentumaik valamint mások által feltöltött dokumentumok között azonos részeket, hasonlóságot, esetleg teljes egyezést keressenek. Az egyetemi diplomabeadási időszakokban a szolgáltatás terhelése igen egyenlőtlen, lökésszerűen akár tízszeresére is felugrik pár napig az átlagos terheléshez képest.

A BonFIRE projekt keretében lehetőségünk nyílt egy részletes kísérlet lefolytatására annak eldöntésére, hogy miként lehet dinamikus cloud (felhő) erőforrások bevonásával a KOPI Plágiumkereső Quality of Service (QoS) paramétereit (mint például a keresési idő) egy adott határon belül tartani ún. elasztikus skálázódással, amely igény szerint dinamikusan csökkenti vagy növeli a szolgáltatás által használt erőforrások számát. A kísérlet egy cloud federációs környezetben zajlik, amely több helyszínen elhelyezett, heterogén cloud hardverekből áll. Ezeknek a lehetőségeknek a kipróbálására és a skálázási paraméterek tesztelésére eddig az éles környezetben nem volt lehetőség. A BonFIRE projekt biztosította a szükséges komplex infrastruktúrát és a monitorozási lehetőségeket a kísérletek elvégzéséhez.

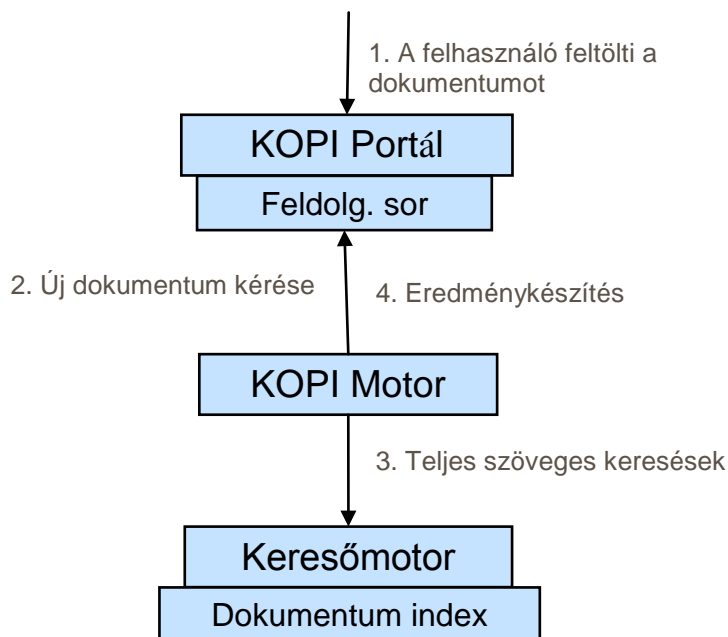
## Bevezetés

A kopi.sztaki.hu portál 2004 óta nyújt egynyelvű plágiumkeresési szolgáltatást, amelyet számos magyar felsőoktatási intézmény rendszeresen használ. 2011-ben a világon elsőként bemutattuk a fordítási plágiumkereső algoritmust [1], amely képes detektálni, ha valaki például az angol Wikipédiából lefordított bekezdéseket használ fel magyar dolgozatában. Az új algoritmus számítási igénye nagyságrendekkel nagyobb, mint az egynyelvű plágiumkeresésé.

A KOPI rendszer felhasználóinak túlnyomó többsége a felsőoktatásban résztvevő hallgató, valamint az ott dolgozó oktatók, ezért a keresések egyenlőtlenül, főleg a diplomaidőszakban érkeznek. Erősen megnő az érdeklődés akkor is, amikor a sajtó felkarolja a plágium ügyét. Az ilyen időszakokban a szerverek terhelése igen egyenlőtlen, lökésszerűen akár tízszeresére is felugrik pár napig az átlagos terheléshez képest.

A cikkben bemutatjuk, hogyan lehet a BonFIRE elosztott felhő tesztkörnyezetet [2] felhasználni terheléelosztási és skálázási kísérletekre a KOPI szolgáltatás számára. A kísérletek az MTA SZTAKI-n belül az Elosztott Rendszerek Osztály és az Informatika Kutatólaboratórium együttműködésében folynak, tervezett befejezési idejük 2013. szeptember.

## A plágiumkeresés folyamata



1. ábra: A plágiumkeresés lépései

A plágiumkeresés során sok különböző rendszerkomponens dolgozik együtt. Érdemes röviden áttekinteni a folyamat lépéseit és azok szereplőit (1. ábra):

- Amikor a felhasználó egy új dokumentumot tölt fel plágiumkeresés végett, a dokumentum a KOPI Portál feldolgozási sorába kerül.
- Az egymástól függetlenül dolgozó KOPI Motorok egyesével dolgozzák fel a dokumentumokat. Ilyenkor a KOPI Motor egy új feladatot kér a Portáltól, és letölti a hozzá tartozó dokumentumot és beállításokat.
- A KOPI Motor az éppen feldolgozott dokumentumot darabolja, a részeket természetes nyelvi feldolgozásnak veti alá, és eközben teljes szöveges kéréseket ad ki a Keresőmotor komponensnek.
- A Keresőmotor a hozzá tartozó Dokumentum indexben elvégzi a kereséseket és visszaküldi a találatokat. A Dokumentum index azon dokumentumokból készül, amelyekben a plagizált szövegrészek eredeti előfordulását keressük.
- A feldolgozás végén a KOPI Motor összesíti az eredményeket és visszaküldi a Portálnak.
- A Portál értesítést küld a felhasználónak, hogy az eredmény elkészült.
- A felhasználó lekéri az eredményt, amely egy listát tartalmaz az esetlegesen másolt részekről és a plágium valószínűségéről.

Egy dokumentum ellenőrzése a dokumentum hosszától függően általában 30-50 percig tart a jelenlegi hardver-környezetben. Ezért a felhasználónak erősen változó időt kell várnia a vizsgálat eredményére, attól függően, hogy mennyi és mekkora dokumentum várakozik az ő dokumentuma előtt a feldolgozási sorban. A szolgáltatás természetéből fakadóan a felhasználás mértéke messze nem egyenletes: a legtöbb ellenőrzendő dokumentum a szorgalmi időszak végén, illetve a diplomamunkák beadásának határideje után érkezik. Tipikus az is, hogy a dokumentumok nem egyesével, hanem nagy csoportokban érkeznek be

feldolgozásra. Amikor túl sok dokumentum érkezik, az eredményekre akár 1-2 napot is kell várni. Ezek a tapasztalatok vezettek minket arra, hogy a KOPI szolgáltatás kapacitását a felhasználói igényeket követve skálázzuk, vagyis nagy terhelés (sok várakozó dokumentum) esetén automatikusan növeljük a feldolgozás kapacitását, és a terhelés csökkenését automatikusan kövessük a felhasznált számítási kapacitás redukálásával. Az ilyen rugalmas skálázás (elastic scaling) megvalósítására a felhő környezetek nagyon jók. Probléma akkor lehet, ha az igényelt kapacitás kinövi a használt felhőt, vagyis nem lehet már több erőforrást adni a feldolgozó folyamatoknak. Ilyenkor alkalmazható a cloud bursting technika, vagyis az addicionális erőforrásokat már egy másik felhőből igényeljük.

Végső célunk az lenne, hogy meg tudjuk becsülni, mikorra készül el egy dokumentum ellenőrzése, és ezt a várakozási időt egy felső korlát alatt tudjuk tartani. Ezáltal tudnánk stabil szolgáltatásminőséget fenntartani a KOPI felhasználói számára.

A kísérlet során tehát valósághűen modellezzük a KOPI működését és a tipikus felhasználói tevékenységet is, vagyis olyan terhelési szituációkat állítunk elő, amelyhez hasonlóak a múltban már előfordultak, vagy amilyenek kezelésére fel szeretnénk készülni. A kísérlet terepül a BonFIRE európai K+F projekt platformját választottuk, és megkaptuk a platform üzemeltetőinek a támogatását is a kísérletekhez, melyek során különféle skálázási módszerek hatékonyságát mérjük heterogén felhő-szövetségekben.

## A BonFIRE tesztkörnyezet

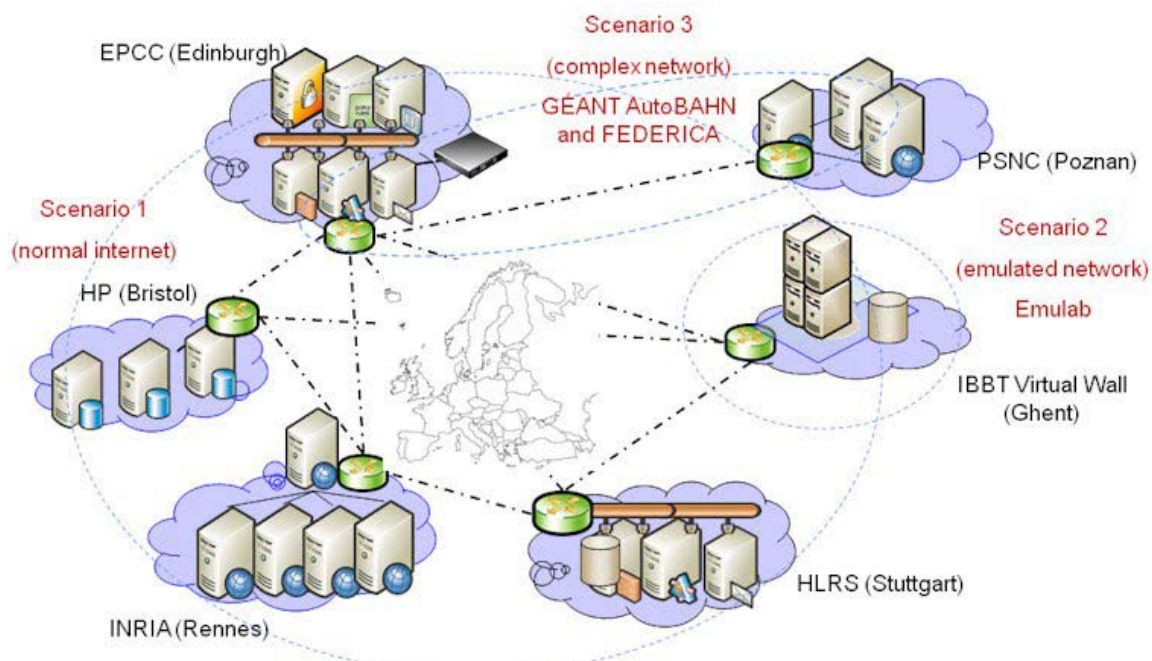
A BonFIRE tesztkörnyezet 6 intézmény saját felhőkörnyezetének egyesítéséből alakult ki. Ez egy decentralizált felhőszövetség, amely egységes interfészt (API) és kezelési lehetőségeket nyújt a szövetségbe integrált 350 processzormagnak és 30 TB háttértárnak. Rövidebb idejű (fél napos) használatra további 3000 processzormag is igényelhető (2. ábra).

A virtuális gépek mérete 8 típusból választható ki, de van olyan lehetőség is, hogy mi adjuk meg a virtuális gép processzormagjainak számát és memóriájának méretét. A virtuális gépek korábban elkészített ún. image-ekkel is indíthatók (ez tartalmazza az operációs rendszert és a szoftvereket), és további adatblokkokat tudunk hozzákapcsolni a géphez, amelyeken az olvasott vagy éppen generált adatokat helyezzük el.

A gépek közötti hálózati kapcsolat területén is sok választási lehetőségünk van. Alapkiépítésben a felhőszövetség összes gépe egy virtuális hálózatban látja egymást, amely a meglévő Internet-kapcsolatokra épül. Ezen kívül lehetőség van dedikált illetve emulált hálózati kapcsolatrendszerek kialakítására az AutoBahn és Virtual Wall szolgáltatásokkal. Ha szükséges, például adott jellemzőjű háttérforgalom generálást is kérhetünk a teszt hálózaton.

A platform használata során ún. kísérleteket hozunk létre, amelyek virtuális gépek, adatok és hálózatok adott konfigurációja. Ezek a kísérletek keretbe foglalják a felhasznált erőforrásokat, és egységesen kezelhetővé teszik azokat. A legkényelmesebb kezelési módszer a BonFIRE portál, ahol a felhasználó böngészheti a kísérleteinek, erőforrásainak az állapotát, és egyszerűbb műveleteket végezhet velük, például virtuális gép indítása vagy leállítása.

Lehetőség van a kísérlet egybevonat megfigyelésére (monitorozására), amelyet egy Zabbix szerveren végezhetünk: minden erőforrásról ide gyűlnek a mérési adatok, amelyeket különféle egyedi grafikonokon ábrázolhatunk. A valódi és virtuális hosztokról is kapunk információt, mint például a számítási terhelés, szabad memória, hálózati forgalom, stb. Ezeket kibővíthetjük a saját szoftvereinket megfigyelő mérésekkel, például a mi kísérletünkben a feldolgozási sor méretét, a feldolgozás sebességét mérjük ezzel a módszerrel.



2. ábra: A BonFIRE tesztkörnyezet

Egy kísérleti kiegészítése a BonFIRE rendszernek a szabályvezérelt skálázódás (Elasticity as a Service, EaaS), amely adott image-ekből készült virtuális gépek számát szabályozza automatikusan. Egy Zabbix mérési kifejezéssel (pl. `{system.cpu.usage.last(0)} > 70`) adhatjuk meg, hogy mikor van szükség új virtuális gép indítására. Ez alapján a rendszer automatikusan elindítja és leállítja a virtuális gépeket amikor szükséges, és elosztja a gépek terhelését HAProxy vagy kamailio terheléselosztón keresztül. Kísérletünkben azonban a támogatottnál bonyolultabb skálázási szabályokat szeretnénk kipróbálni, ezért az EaaS a mi esetünkben nem alkalmazható.

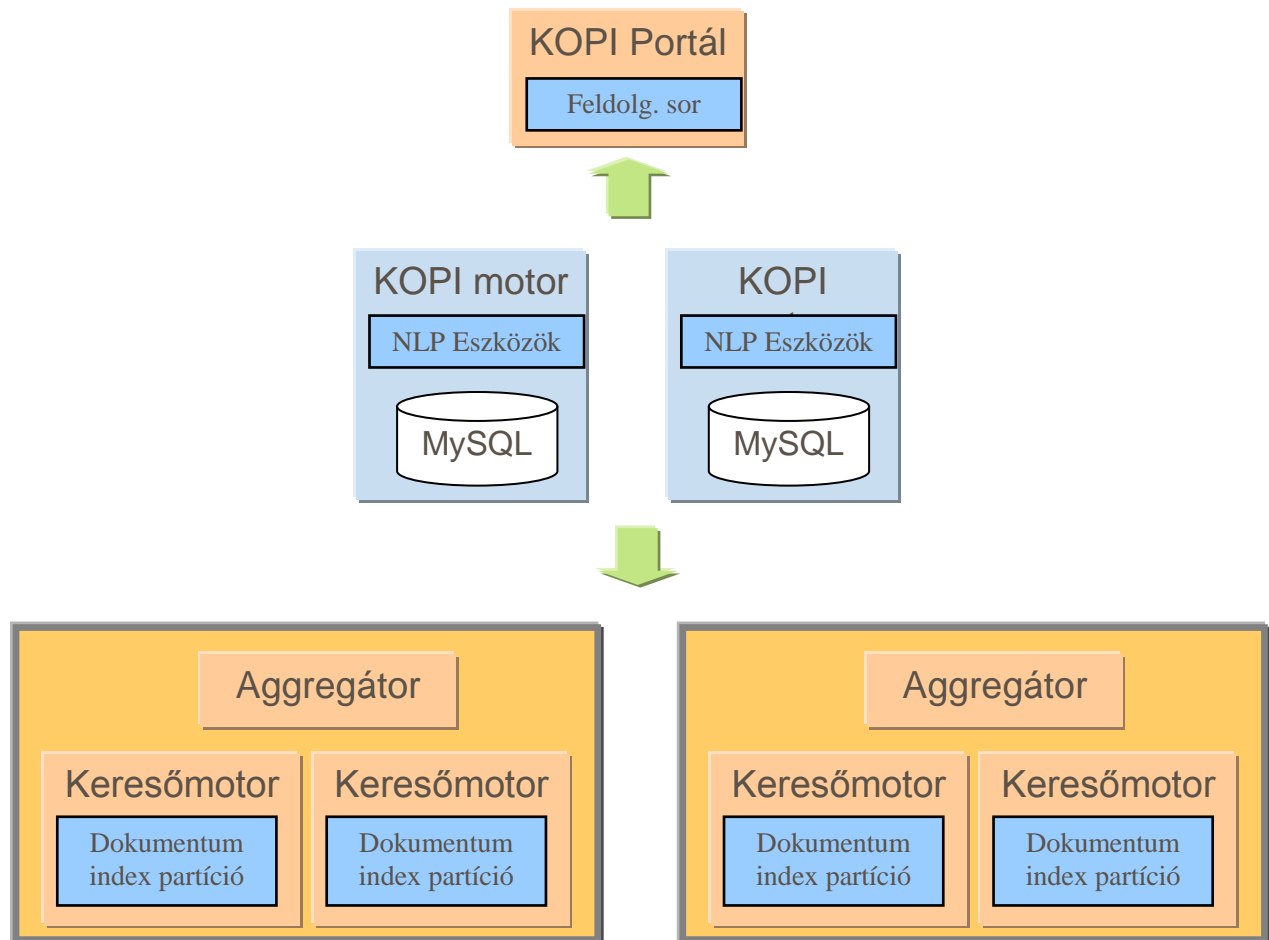
Az automatikus értesítési rendszer használatával a kísérlet állapotváltozásairól kaphatunk üzeneteket, például virtuális gépek elindulásáról, leállításáról, stb. Ezt a funkciót a RabbitMQ valósítja meg.

A BonFIRE 2013 elején meghirdette a nyílt hozzáférést, vagyis a tesztkörnyezet használatára jelentkezni lehet a <http://www.bonfire-project.eu/involved> webcímen, és az elfogadott kísérletjavaslatok 4 hónap ingyenes hozzáférést kapnak a BonFIRE szolgáltatásokhoz. A tesztkörnyezet működésének fenntartását 2014 ősziéig vállalta a BonFIRE projekt.

## A KOPFire kísérletek

Egy elosztott rendszer skálázhatósága lehet vertikális és horizontális irányú: míg az első esetben meglévő erőforrásaink (*ez lehet virtuális gép, tárterület, stb.*) kapacitását növeljük, horizontális skálázásnál újabb és újabb erőforrásokat vonunk be a rendszerbe, ezáltal növelve alkalmazásunk elosztottságát. Cloud rendszerekre épülő szolgáltatások esetén az utóbbi a könnyebb és gyakoribb skálázási „irány”, mivel a virtuális gépek konfigurációjának megváltoztatása gyakran a futó komponensek újraindítását igényli. A BonFIRE projekt azonban az egy cloud rendszeren belüli skálázhatóság tesztelése mellett kiemelt szerepet szán a cloud federációk közötti terheléselosztás lehetőségeinek vizsgálatára is. Kritikus időszakokban, amikor a terhelés hirtelen megnő, előfordulhat, hogy más cloud szolgáltatótól is bérelnünk kell erőforrásokat. Habár a több különböző cloud-ba történő skálázódás különösen kecsegtető lehet akkor, amikor alkalmazásunkat földrajzi értelemben is elosztottá

kívánjuk tenni (például a felhasználóink kéréseit mindig a földrajzi értelemben vett legközelebbi adatközpontba irányíthatjuk), rejt azonban néhány nem egyszerűen megoldható problémát is. Különösen nehéz kérdés az adatok cloud-ok közötti szinkronban tartásának kérdése.



3. ábra: A KOPI elosztott architektúrája

A KOPI szolgáltatás „felhősítése” során több lehetőség is felmerült a skálázhatóság vizsgálatára, esetünkben azonban a legfontosabb a hirtelen jelentkező (például diplomamunka-leadás időszakában) felhasználói aktivitásra történő felkészülés volt. A KOPFire kísérletek során egy olyan időszakot próbálunk szimulálni, amely során a terhelés váratlanul a sokszorosára nő. A rendszer, automatikusan észlelve a terhelésben fellépő változást, az egyes komponensekből újabb és újabb példányokat indítva próbálja a feladatokat egyenletesen elosztani, egészen addig, amíg az átlagos kiszolgálási idő meg nem közelíti a kísérlet elején meghatározott értéket. Egy másik kísérlet során a skálázás nem virtuális gépek, hanem cloud szinten történik: a teljes rendszer annak összes komponensével klónozásra kerül egy másik cloud szolgáltatásban. A két egymástól függetlenül futó szolgáltatást egy terhelélosztó alkalmazás köti össze. Mindkét kísérlet a mesterséges terhelés hirtelen csökkentésével zárul, így ezzel vizsgálhatóvá válik a rendszer elasztikussága, azaz hogy képes-e az ideiglenesen igényelt extra erőforrások gyors elengedésére is.

A fentiekkel összhangban a kísérletek során nem foglalkoztunk a dokumentumok indexelésével, mivel azokat az MTA SZTAKI Hadoop[3] clusterén előre legyártottuk, majd az adatfájlokat felmásoltuk a BonFIRE cloud-okon előre lefoglalt adatblokkokra. A

skálázhatósági kísérletek folyamán kizárólag az egy dokumentum átlagos feldolgozási idejének mérésére koncentrálnak.

A 3. ábra mutatja be a KOPI szolgáltatás elosztott változatának architektúráját. Növekedő terhelés esetén az egyes komponensek számát növeljük, bizonyos komponensek azonban nagyon egyszerű feladatot látnak el, így azok párhuzamosítására nincs szükség.

A KOPI Portál tartja a kapcsolatot a felhasználókkal, itt lehet dokumentumokat feltölteni, és innen tudnak a feldolgozóegységek dokumentumokat kérni.

A KOPI Motor végzi a dokumentumokban a plágiumellenőrzést, egyszerre egy dokumentummal foglalkozik, és közben számos keresőkérdést tesz fel a keresőklaszternek. A KOPI Motor viszonylag egyszerűen skálázható, mivel egy virtuális gépet kell csak indítani új motor létrehozásához. Ez történhet a helyi, vagy távoli felhőben is, amennyiben a megfelelő image rendelkezésre áll a távoli felhőben is. Ellenkező esetben az image áttöltése a másik felhőbe jelentősen megnöveli a motor indítási idejét.

A Keresőklaszter három fontos komponensből áll: az Aggregátor, a Keresőmotor és a Dokumentumindex.

Az MTA SZTAKI által fejlesztett szabadszavas Keresőmotor a KOPI Motortól érkező kérdésekre keres rá a Dokumentumindexben. Ebből a komponensből több példány fut párhuzamosan, és mindegyik példány a dokumentumok egy-egy részhalmazában keres. Egy keresési feladat tehát párhuzamosan hajtódik végre, és a dokumentum-partíciók számának növelésével az átlagos keresési idő csökkenthető. A Keresőmotor a legnagyobb erőforrás-igényű komponens, legalább 2-4 CPU magra és 8-16 GB memóriára van szüksége.

Az Aggregátor egy igazán könnyűsúlyú komponens, amely hidat képez a KOPI Motor és a Keresőmotorok között. A KOPI Motortól érkező kéréseket leküldi az összes Keresőmotor példánynak, majd a tőlük érkező találati listákat összefűzi és relevancia szerint rendezve visszaküldi. Az Aggregátor komponensből egy példány futtatására van szükség keresőklaszterenként.

A Keresőmotor alatt lévő Dokumentumindexben a dokumentumok fix módon kerültek partícionálásra, így minden Keresőmotor saját index-partícióval rendelkezik. Skálázódáskor a partíciók darabjainak – és így a klaszterben futó Keresőmotor-példányok – számát nem tudjuk tetszőlegesen növelni, mivel az a teljes index újradarabolását igényelné. Ezért a skálázást klaszter szinten képzeljük el: vagy egy egész új klasztert indítunk, vagy a klaszter keresőmotorjainak teljesítményét emeljük meg. Ez esetben is lehetséges új klasztert másik felhőben indítani, de az index nagy mérete és a felhők közötti átlagos Internet-kapcsolat sebessége miatt itt mindenképpen előre át kell tölteni a teljes dokumentumindexet a másik felhőbe.

A skálázási műveletek automatikus végrehajtására három megoldást is ad a BonFIRE platform. Az első egy API felület, amellyel REST módon lekérdezhethetjük és módosíthatjuk erőforrásaink állapotát. Ez az API az OCCI (Open Cloud Computing Interface [4]) egy bővített változata, és XML vagy JSON formátumú erőforrás-leírók fel- és letöltésén alapul a működése. Mivel ezeket a leírókat kicsit nehézkes előállítani, ezért parancssorban végrehajtható szkriptek is készültek, melyek a szokásos shell-scripting módszereivel alkalmazhatók. A harmadik lehetőség egy Ruby nyelven íródott programkönyvtár, amellyel Ruby objektumokon keresztül vezérelhetjük erőforrásainkat.

# Összefoglalás

Az MTA SZTAKI a BonFIRE kísérleti környezetben több felhőből álló rendszerekben, úgynevezett felhőszövetségekben kísérleteket hajtott végre, amely során az alkalmazásai által felhasznált hardveres erőforrások mennyiségét növeli illetve csökkenti, miközben folyamatosan naplózza a teljesítményben fellépő változásokat. A résztvevők ezáltal értékes tapasztalatokat gyűjthetnek arról, hogy alkalmazásuk az adott felépítés mellett mennyire skálázható, illetve kitapasztalhatják, hogy adott terhelés mellett mennyi erőforrás szükséges a megfelelő kiszolgálási sebességhez. A terhelés alapján az optimális konfigurációt megadó számítások birtokában az MTA SZTAKI a szolgáltatásainak minőségi kritériumait szeretné jobban becsülhetővé tenni, illetve erőforrásainak felhasználását optimalizálni.

A tesztelési, kipróbálási lehetőségek igen korlátozottak a felhőszolgáltatások területén, különösen, ha felhőszövetségekről van szó, ezért van kiemelt jelentősége a BonFIRE tesztkörnyezetnek, amely ingyenes hozzáférési periódust hirdetett meg 2014 őszéig. A KOPFire kísérlet nem csak a KOPI Plágiumkeresőnek hasznos, hanem hozzájárul a BonFIRE projekt céljaihoz is azáltal, hogy az ott használt monitorozási eszközöket értékeli, és kipróbál különböző QoS paraméterezési stratégiákat, amelyből más szolgáltatások is profitálhatnak.

## Hivatkozások

- [1] M. Pataki, "A new approach for searching translated plagiarism", 5th International Plagiarism Conference, 16-18 July 2012, Newcastle
- [2] BonFIRE projekt honlap, <http://www.bonfire-project.eu/>
- [3] Apache Hadoop, <http://hadoop.apache.org/>
- [4] OGF Open Cloud Computing Interface Working Group, <http://occi-wg.org/>