Acta Agraria Kaposváriensis (2011) Vol 15 No 3, 93-102
Kaposvári Egyetem, Állattudományi Kar, Kaposvár
Kaposvár University, Faculty of Animal Science, Kaposvár

Szűk keresztmetszetek feltárása informatikai rendszerekben

Boros J.
Nyugat-magyarországi Egyetem, Faipari Mérnöki Kar, Informatikai és Gazdasági Intézet, 9400 Sopron, Bajcsy-Zsilinszky u. 9.

ÖSSZEFOGLALÁS

Napjainkban a hétköznapi életünkben is egyre inkább helyet követelnek maguknak az informatikai megoldások, kisebb-nagyobb hálózatok, az Internet, a web. Folyamatosan jelennek meg az új, átfogó, robosztus rendszerek, weben működő alkalmazások. Manapság egy korral lépést tartó cég működésének elengedhetetlen velejárója az informatika, az interneten történő megjelenés. Növekvő teret nyer a hálózaton keresztül történő ügyintézés, vásárlás, dokumentumkezelés, de ide sorolható a termelő munkát elősegítő informatikai tevékenység is. A fejlesztések, kutatások során sok alkalmazás - amely eddig „elzártan", egy asztali számítógépen futott - következő verziója már weben, hálózaton keresztül érhető el. A hálózatok működtetésének nagyon fontos szempontja a megbízhatóság, rendelkezésre állás, amelyet minden esetben úgy célszerű biztosítani, hogy ne méretezzük túl az adott rendszert. Ugyanakkor figyelembe kell venni azt is, hogy sok rendszer esetén tapasztalható rövid ideig tartó, de a környezethez képest kiugróan magas igénybevétel. Elsődleges feladatunknak tekintjük ezért a rendszerek szűk keresztmetszetének feltárását, és megoldási javaslatok publikálását általánosan és konkrét hálózatok, informatikai rendszerek esetén is. Jelen cikkben egy tesztkörnyezetben keressük folyamatok szintjén, hogy az egyes szálak mekkora terhet rónak a PC-re, azaz mennyi memóriát, CPU időt használnak és mekkora az írás illetve olvasás egy-egy folyamatot tekintve. A mért értékekből statisztikát generálunk , majd a kritikus értékek környezetét elemezve olyan, későbbiekben is felismerhető helyzeteket fogalmazunk meg, melyek esetén a rendszer túlterhelése várható.
(Kulcsszavak: szűk keresztmetszet, informatika, rendszer)

ABSTRACT

Finding bottlenecks in IT systems

J. Boros
University of West Hungary, Faculty of Wood Sciences, Institute of Informatics and Economics
H-9400 Sopron, Bajcsy-Zsilinszky u. 9.

Nowadays, more and more segments of the information technology (such as the networks, the Internet, the World Wide Web) become essential in the everyday life. We get new IT-systems that are often robust and comprehensive. Companies need to use information systems to keep abreast of the times. In the past information systems have appeared stand-alone, but now they often use networking solutions for office work, document management etc. Although the reliability and availability are very important aspects of the services the efficient provision of these attributes with the smallest hardware, software and money investment is also crucial. IT systems are dimensioned in that manner that their load is light or medium in most of their life, however, they suffer heavy load in a small fraction of their operation. Since this heavy load can cause critical situations our main goal is to

explore and identify the bottlenecks of systems, and to give proposals how to avoid the problems. We use a test environment where we measure the load factors of each threads or processes. Our measurements cover memory usage, CPU usage and I/O data traffic. Based on the observed data and their operation environment, tendencies are identified that help system operators to prevent systems from overloaded situations.
(Keywords: bottlenecks, information technology, system)

BEVEZETÉS

Az informatika területén, mint sok más területen mindennek megtalálható az oka, minden rendszerbeli esemény visszavezethető valamire. Ez kisméretű rendszerek, vagy egy könnyen átlátható normál asztali PC esetén viszonylag egyszerűen kideríthető. A nagyméretű, bonyolult felépítésű szerverek vagy sok esetben szerverfarmok esetén azonban nem triviális meghatározni azt, hogy mi miért történt. Nézzünk egy egyszerű példát: a Felvi oktatási rendszere egész évben megfelelően működik, minden hiba és gond nélkül. A jelentkezési határidők környékén azonban gyakran volt tapasztalható olyan anomália, amikor a rendszer elérhetetlenné vált. Célunk, hogy egy ilyen rendszer előzetes megvizsgálása után meg tudjuk mondani, hogy milyen módon lehet minimális anyagi ráfordítás mellett hirtelen megugró terhelés esetén is teljesíteni a követelményeket (*Eckert és mtsai.*, 2007).

Fontos megemlíteni, hogy nem tisztán kutatásról van szó, hanem az eredmények gyakorlati hasznosításra is kerülnek. Célunk egyértelmű: egy tetszőleges informatikai rendszer szűk keresztmetszeteit kívánjuk meghatározni. A szűk keresztmetszetek alapján megmondható, hogy az adott rendszer miért omlik össze bizonyos terhelés hatására, és így tervezhetővé válik az informatikai beruházás. A kérdés fontosságát mutatja, hogy alapvetően két típusú fejlesztést értelmezhetünk. Vertikális (függőleges) fejlesztés esetén a meglévő hardvereszközeinket cseréljük le erősebb, nagyobb terhelést bíró eszközre. Tipikus példa lehet, ha processzort cseréljük 2.0 GHz órajelűről 3 GHz órajelű egységre. Ezzel szemben a horizontális bővítés annyit takar, hogy több, de egyenként kisebb teljesítményű hardvert állítunk szolgálatba. Ekkor szerverek láncolatával (más néven szerverfarmmal) biztosítjuk a szükséges erőforrást. Ebben az esetben azonban külön figyelni kell a felmerülő szinkronizációs feladatok megoldására is (*Menascé és Almeida*, 2002).

A vertikális megoldás viszonylag olcsóbb, de a korlátait is hamarabb eléri, mint a horizontális fejlesztés. Nyilvánvaló, hogy a rendelkezésre álló technológiai korlátokat nem tudjuk átlépni, azaz hiába hoztuk ki a vizsgálat alapján azt az eredményt, hogy egy többmagos, nagy teljesítményű órajellel rendelkező processzor elegendő lenne, nem tudunk beszerezni a piacon ilyen erősségű processzort. Másik oldalról nézve, a horizontális drágább megoldás, cserébe itt közvetlenül „nincs” határ, de a költségek között ekkor megjelennek a különböző szinkronizációs és egyéb kiadások is.

A kutatás irányát a budapesti székhelyű Netvisor Zrt. piaci igényei és szakmai tapasztalatai határozták meg.

ANYAG ÉS MÓDSZER

Folyamat

A 2010 tavaszától, 2011 végéig terjedő K+F folyamatot négy nagyobb, jól elkülöníthető részre bontottuk. Jelenleg az első és második szakasz már lezárult, pillanatnyilag a harmadik ponttal foglalkozunk.

1. szakasz
Az első feladatot a felhasználható eszközök feltérképezése jelentette. A Unix (Linux) rendszerek működési és vizsgálati szempontú megismerését a további kutatási irányok kijelölése követte (*Boros és mtsai.*, 2010).

2. szakasz
A második szakaszban egy modult fejlesztettünk ki egy meglévő rendszer alá. A PVSR mérőmodul fejlesztést meghatározza, hogy a piaci partner – érthetően – a saját rendszerét kívánja használni a különböző szerverek vizsgálata során. Ez természetesen korlátokat jelent a szóba jövő megoldások kiválasztásában, ugyanakkor egyértelműen kedvező abból a szempontból, hogy a kialakított rendszer éles alkalmazásra is kerülhet valóságos alkalmazások monitorozásában (*Jereb és mtsai.*, 2011).

3. szakasz
A következő lépést egy nagy rendszer és annak kiszolgáló szervereinek monitorozása jelenti a létrehozott modullal. Ez lehet piaci szolgáltatás, például az adóhivatal interneten elérhető rendszere, de lehet az oktatás területén használatos szolgáltatás is, mint például a Neptun rendszer. A vizsgálat során a mért adatok elemzésével megkaphatjuk, hogy mely eszközök mely paramétereinek változása volt kritikus a rendszer működésére nézve. Így meg tudjuk határozni a szűk keresztmetszeteket egy rendszeren.

4. szakasz
Végül a negyedik, befejező feladat lényege, hogy a 3. szakaszban végzett vizsgálat alapján meghatározzuk, mit lehet kezdeni a szűk keresztmetszetekkel, azaz mit kell cserélni, bővíteni, javítani ahhoz, hogy a rendszer kezelni tudja a kritikus időszakok nagyobb terhelését is. Nyilvánvalóan egy adott hardvernek vannak olyan paraméterei, amit nem lehet bővíteni vagy cserélni. Ebben az esetben azonban még mindig költséghatékonyabb kicserélni az adott fizikai eszközt (pl. alaplapot), mint egy nagyobb méretű informatikai beruházást végrehajtani.

PerformanceVisor (PVSR)
A PerformanceVisor (PVSR) egy Solaris-Windows-Linux szerver-platformon futó Web-alapú alkalmazás, amely különböző – például hálózati vagy IT – eszközök és alkalmazások monitorozását teszi lehetővé. Sokrétű, elosztott monitorozó rendszer, elosztott mérő szerver struktúrával, mérő szerver csoportok kezelésével, azon belül automatikus terhelés-megosztással. A rendszer képes öntesztelésre, így az esetleges belső hibák feltárására. Egyszerűen kiterjeszthető mérőprotokollal rendelkezik, így könnyen fejleszthető hozzá új, általános vagy akár speciális igényt kielégítő mérő eszköz. Dokumentált API-val (programozói interfésszel) rendelkezik, ezért alkalmazásával általános mérő szerver-keretrendszer is egyszerűen implementálható.

A mérési intervallumok választhatók: a 15 másodperctől egészen az órás ciklusig, a létrehozásnál kiválasztott intervallumot a felhasználó a későbbiek során akár módosíthatja is. A távoli mért eszközöket elég, ha a mérő szerverek érik el, a kiolvasott adatokat biztonságos úton (SSH és SFTP protokollok segítségével) továbbítják a központi gép és a felhasználói felület felé. A mérő szervereknek csupán az elinduláshoz van szükségük a központi Oracle szerverre, ennek köszönhetően akár egy Oracle adatbázis leállás sem eredményez mérés kiesést. Az adatbázis kezelő egy korszerű, piacvezető eszköz mind a konfiguráció, mind a mérési adatok tárolására.

A PVSR nagyon sok funkciót támogat, megkönnyítve a munkát. Felületről mérésenként szabályozható adatmegtartás (data retention) és tömörítés, telepítéskor választható partícionált vagy nem partícionált adatbázis használattal rendelkezik a

rendszer. A PVSR Webservice SOAP interfésze egy standard felületet nyújt, amin keresztül mind a konfiguráció mind a mérések és riasztások lekérdezése is megvalósítható, így az alkalmazás könnyen kapcsolható más társrendszerekhez és biztosítja az átjárhatóságot. A rendszert az egyszerű konfigurálhatóság jellemzi. Így a sablonokkal segített objektum létrehozás: eszközök, riasztások, grafikonok és mérések automatikus felderítése az eszközökön és szűrhető listában történő megjelenítése is elérhető szolgáltatás.

Amit fontos kiemelni, hogy a rendszer a kapott adatokból automatikusan, jól használható és átlátható grafikonokat képes létrehozni. Ezek flash alapú grafikonok, amelyek gazdag kliens oldali interaktivitást tesznek lehetővé: nagyítást, minimum/maximum beállítását, elemek elrejtését/megjelenítését, trend vonalak felvételét. A kialakított mérések tetszőleges köre összevonható egy darab publikus és/vagy privát grafikonra, sőt a mérések értéke tetszőleges kifejezéssel feldolgozható és ábrázolható (például két mérés összege, különbsége, stb).

Lehetőség van több eszköz méréseit összegyűjteni egy oldalra, majd az így kapott egyedi lekérdezést a felhasználó rögtön el is mentheti virtuális eszközként, a grafikonokon tetszőleges időponthoz tartozó mérési görbe megjeleníthető. Széleskörűen megfogalmazhatunk küszbérték feltételeket, amelyek megsértése esetén az alkalmazás riasztást generál levél és/vagy SNMP trap és/vagy tetszőleges parancs formájában. Feltételként fix érték vagy a korábbi mérések alapján számított átlag és szórásra támaszkodó érték adható meg. A feltételben nem csak az aktuális mérési értéket lehet megadni, hanem trend számítás alapján kalkulált jövőbeni értékeket is (például diszk kihasználtság egy hét múlva).

A jogosultsági szintek is nagyon rugalmasan használhatóak. A rendszerben ötszintű felhasználói rendszer található, ahol egyedileg szabályozható, hogy melyik felhasználó melyik telephelyet-eszközt-mérést-riportot-grafikát tekintheti meg, illetve hogy láthatja-e a threshold sértéseket. Természetesen egy adott telephely vagy eszköz riport nézetének megtekintéséhez mind a riport mind az objektum elérhető kell hogy legyen a felhasználó számára.

Linux eszközök

A Linux operációs rendszer teljesítmény elemzéséhez olyan standard és speciális (egyedi, kísérleti) eszközöket keresünk, amelyek pontos és hasznos információkat szolgáltatnak a Linux kernel felett futó folyamatok memória, CPU, diszk terhelési adatairól, és ezek alapján összefoglaló képet lehet kapni a rendszer állapotáról.

Az Unix rendszer a működési információkat a /proc fájlrendszer alá helyezi. Ez egy speciális fájlrendszer, amely a diszken nem jelenik meg, a memóriában kerül lefoglalásra. A /proc alatt mind a rendszer hardveres, mind pedig a szoftveres működési paraméterei naplózásra kerülnek különböző fájlokba, amelyek aztán kiolvashatók.

A projekt 1. szakaszában a standardnak számító Vmstat, Iostat, Top, Ps, stb. monitorozó programok képességeit és a rendszeranalízis szempontjából hasznos szolgáltatásait tekintettük át, továbbá olyan opensource eszközöket kerestünk, amelyek a fentieken túl további hasznos információkat tudnak nyújtani. Megállapítható volt, hogy mindegyik mérési program a /proc fájlrendszer alapján dolgozik. Így a különbségük azoktól a tényezőktől függ, hogy milyen paramétereket tartanak fontosnak a programozók és milyen ügyesen valósították meg a programjukat.

Esetünkben fontos szempont, hogy a későbbi elemzés céljából egy adott eszköz képes-e folyamatos megfigyelésre és elmenthetők-e az idősor adatokat diszkre. Ezen

szempont figyelembevételével módszert dolgoztunk ki a különböző mérő eszközök összehasonlítására.

A mérések és tapasztalatok alapján a dstat széles körben és könnyen használható, mivel segítségével egy adott rendszeren könnyen lehet mérni CPU, memória, I/O és egyéb paraméter változásokat. Ugyanakkor alkalmazását nagy mértékben korlátozza, hogy használatához különböző segédprogramok és egyéb mérést segítő eszközök telepítésére van szükség. Ez sok esetben a valós használat során nem tehető meg, így a dstat a projekt célkitűzései szerinti környezetben nem használható.

A második vizsgált eszköz az iotop volt. Az iotop alapesetben az összes olvasott illetve írt adat mennyiségét szolgáltatja a rendszer egészét, azaz az összes lemezt – és persze folyamatot – tekintve. Ennek során megkapjuk a folyamat azonosítóját (PID), a folyamatot futtató felhasználót (USER), folyamatonkénti bontásban az írt és olvasott adatmennyiséget byte/s-ban, továbbá megtudjuk, milyen arányban foglalkozott a folyamat (szál) swap műveletekkel illetve milyen arányban várt a folyamat (szál) I/O műveletekre. A parancs folyamatosan fut, mód van futási időben változtatni néhány jellemzőt. Adott időközönként frissíti a listát a folyamatokról, amivel értesülünk a rendszer változásairól.

A harmadik tételesen megvizsgált eszköz az iostat volt, amely az előzőeknél némileg szofisztikáltabb eszköz. Segítségével CPU vagy diszk műveleteket is tudunk monitorozni. A diszkre koncentrálva megállapítható, hogy könnyen paraméterezhető, így egyszerűen lehet a riportot szűkíteni a számunkra fontos eszközökre. Ez az eszköz alapvetően nem a folyamatokkal, hanem az eszközökkel foglalkozik, így azonosítja a gépet, az operációs rendszer verzióját, a processzorok számát, típusát. Egy eszközhöz megadja a másodpercenkénti transzferek számát, az olvasott és írt adatok mennyiségét (paramétertől függően blokkban, kilobyte-ban vagy megabyte-ban). Alapértelmezésben egyszeri mintavételezésről van szó. Ha azt szeretnénk, hogy több mérés történjen, akkor ez paraméterezéssel biztosítható.

Az utóbbi két eszközt összehasonlítva elmondható, hogy az iotop inkább a processzenkénti bontásra koncentrál, és nehezebb összegző adatokat kinyerni segítségével a rendszer egészéről. Az iostat ezzel szemben az eszközönkénti terhelést tükrözi. Kényelmi funkciói többletet jelentenek az iotop-hoz képest (pl. időbélyeg készítés), ugyanakkor a processzenkénti vizsgálatot inkább az iotop eszközzel lehet eredményesen végezni, és az esetlegesen fontos SWAP paraméterekről is inkább az iotop tud információval szolgálni.

Végrehajtottunk egy olyan méréssorozatot is, amelyben mindhárom eszközzel, ugyanazon terhelés mellett, azonos környezetben mértünk. Bár az egyes mérőalkalmazások más-más szemszögből mértek, az összehasonlítás során azt tapasztaltuk, hogy a kapott értékek hasonlóak, azaz az alkalmazás működési tendenciái nyomon követhetők.

Összefoglalva megállapítható, hogy a három eszköz eltérő alkalmazásokra készült, és bár mindegyik eszköz nagyon hasznos és jól működik a saját területén, de viszonylag speciális követelményeink kielégítésére saját fejlesztésű környezetet célszerű létrehozni. A három mérőalkalmazás tapasztalatainak felhasználásával egy saját szkriptet hoztunk létre, amely már megfelel a rendszer és a mi elvárásainknak is.

EREDMÉNY ÉS ÉRTÉKELÉS

Az eddigi fejlesztések eredménye alapján az általunk fejlesztett mérőszerver képes szolgáltatni különböző informatikai rendszerek (szerverek) vizsgálata után mért értékeket, amiket a PVSR segítségével meg tudunk jeleníteni grafikonon illetve a mögötte található adatbázisban képesek vagyunk tárolni azokat. Ez a további, elemzési

lépések szempontjából fontos, így biztosítva van az adatok konzisztens, minden igényt kielégítő tárolása.

A következőkben néhány példát láthatunk a működésről illetve a grafikonokról a mért értékekkel. A grafikonokon egy Linux rendszert monitoroztunk, amely az alábbi paraméterekkel bír: AMD Athlon CPU 1.7 Ghz órajel, IDE 120 Gb diszk, 1.5 Gb swap, 750 Mb RAM memória, x86-os CentOS 5.5 Linux rendszer.

Az *1. ábrán* látható az Oracle processzekre illesztett CPU mérés által szolgáltatott eredmény. Mint a grafikonról is leolvasható, a CPU terhelés rövid időn belül is képes nagyon eltérő értékeket szolgáltatni, függően természetesen az egész rendszer viselkedésétől. Ennek „párja" a *2. ábrán* látható értéksorozat, amely ugyancsak a CPU terhelését mutatja, de azzal a különbséggel, hogy itt nem a „sys" terhelés, hanem a folyamatok „usr" terhelése látható.

1. ábra

CPU sys terhelési értékek oracle processzekre illesztve

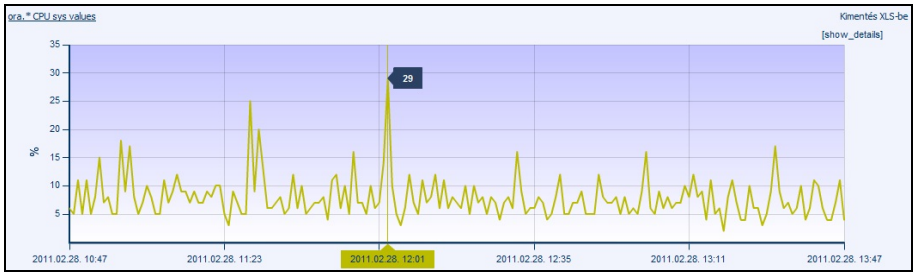

Figure 1: CPU sys load factor on oracle processes

2. ábra

CPU usr terhelési értékek oracle processzekre illesztve

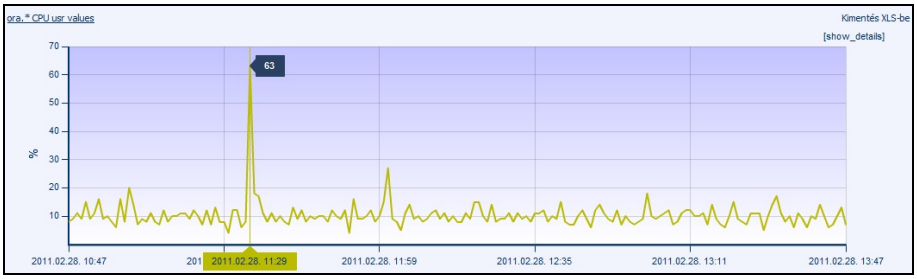

Figure 2: CPU usr load factor on oracle processes

A *3.* és *4. ábrán* a memória terhelés követhető nyomon. Hasonlóan a CPU terheléshez itt is kettéválik a két monitorozott érték, a folyamatok fizikai memória használata illetve a swappelt, azaz közösen használt értékek.

3. ábra

Fizikai memória terhelési értékek oracle processzekre illesztve

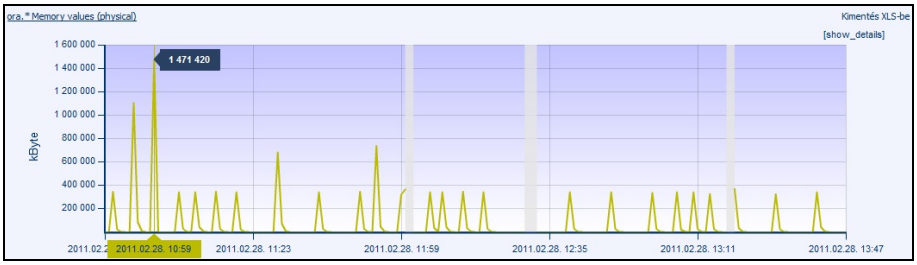

Figure 3: Physical memory load factor on oracle processes

4. ábra

SWAP memória terhelési értékek oracle processzekre illesztve

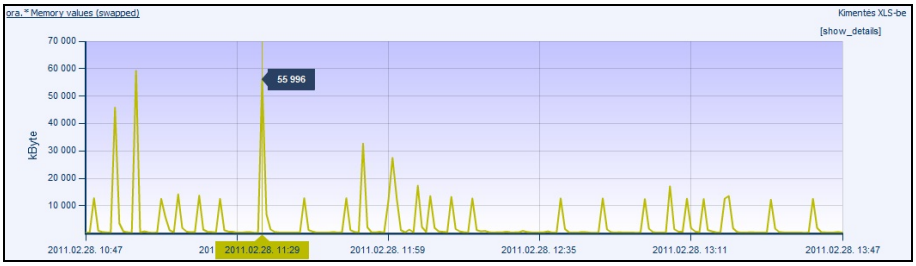

Figure 4: Swapped memory load factor on oracle processes

A swap értéknél fontos megjegyezni, hogy a különböző folyamatok által közösen használt területek (átfedések) nem jelennek meg különállóan. Másik oldalról közelítve, a közös területek minden folyamatnál megjelennek, így többszörösen kerülnek megjelenítésre. Szélsőséges esetben így nagyobb értéket kaphatunk egy összegzés után, mint a fizikai memória mérete az adott rendszerben.

A harmadik nagyobb vizsgált értékcsoport az I/O értékek. A futás során minden folyamat rendelkezik olvasott illetve kiírt adatmennyiséggel. Ez a különböző háttértároló eszközök vizsgálata esetén lehet érdekes, mivel tudjuk, hogy egy rendszer esetén az I/O műveletek a leglassabb részfolyamatok. Szintén két külön grafikon található, egyrészt az írási (*5. ábra*) másrészt az olvasási (*6. ábra*) értékek vizsgálati eredményeinek prezentálására.

Mint korábban is olvasható, a fenti értékek valamilyen összegzett számot takarnak egy adott pillanatra nézve is. Egyetlen esetben jelentik a tényleges folyamat általi terhelést, mégpedig ha csak egyetlen folyamatra illeszkedik a mérés indexe (a fenti példákon az ora* karaktersorozat). Ahhoz, hogy további számítások történhessenek megközelítve a tényleges folyamat szintű terhelést, fontos tudni, hogy adott pillanatban a kapott terhelés hány folyamat között oszlik meg, ezt is külön grafikonon (külön mérés definícióként) tudjuk kezelni (*7. ábra*).

5. ábra

Olvasott adatmennyiség oracle processzekre illesztve

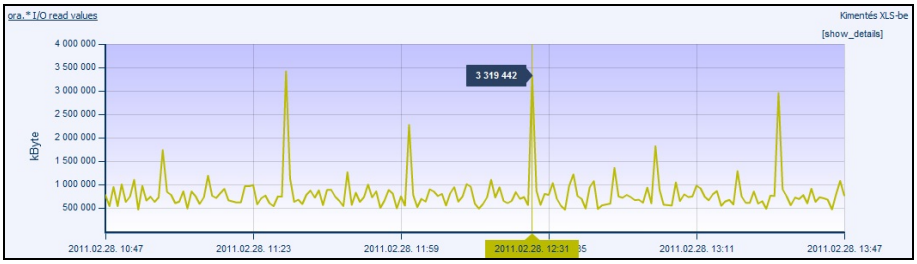

Figure 5: Read data by oracle processes

6. ábra

Írt adatmennyiség oracle processzekre illesztve

Figure 6: Wrote data by oracle processes

7. ábra

Oracle processzek száma az idő függvényében

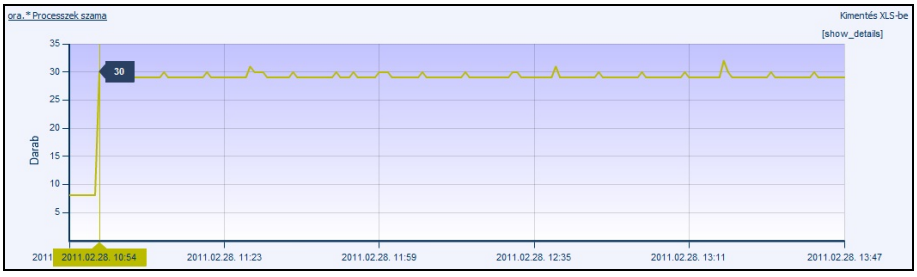

Figure 7: Number of oracle processes

KÖVETKEZTETÉSEK

Az elmúlt időszakra tervezett célokat sikerült elérni. Elkészült egy PVSR alatt használható mérőszerver, melynek segítségével adatokat kaphatunk egy tetszőleges Linux rendszerről. A megvalósítás során nem használtunk harmadik fél által szolgáltatott illetve készített eszközt, mérő algoritmust, az adatok átmeneti tárolását illetve feldolgozását is saját fejlesztésű program hajtja végre. Az elkészült eszköz tetszőleges folyamat monitorozását hajtja végre szolgáltatva róla CPU, memória és I/O értékeket. Az eszköz segítségével összesített eredmények kerülnek a PVSR rendszerbe és így a grafikonokra is. Lehetőség van a folyamatok számát is nyomon követni, így átlagot vagy egyéb számított eredményeket is lehet generálni.

Az eddigi eredmények - a tesztrendszerek vizsgálata alapján is - bíztatóak. A mért eredmények segítségével az adott folyamatok viselkedése meghatározható, így képesek lehetünk a különböző tendenciák, analógiák felismerésére. Ezeket a viselkedési mintákat a későbbiekben sok esetben használhatjuk. Részleteiben feltérképezve egy széles körben használt és nagy terhelést generáló programot, rámutathatunk annak ismétlődő viselkedésére, kritikus pontjaira. Így egy még nem vizsgált rendszer várható viselkedésére vonatkozóan is adhatunk olyan információkat, amelyekkel a későbbi kritikus helyzetek esélyét jelentősen csökkenthetjük.

Továbblépési lehetőségek

A kialakított monitorozó keretrendszer természetesen folyamatos tesztelést igényel különböző alkalmazási környezetekben. Ennek fontos részét jelenti a Netvisor Zrt. környezetében történő „éles” tesztelés, az eszköz működésének validálása.

Miután elkészült egy validált, minden igényt kielégítő eszköz, egy valós rendszert vagy annak reprodukcióját lehetne monitorozni, vizsgálni. Az így kapott adatok alapján lehetőség van az adott rendszer szűk keresztmetszeteinek feltárására, modell felépítésére és megoldás szolgáltatására.

KÖSZÖNETNYILVÁNÍTÁS

Az eredmények elérésében meghatározó volt a Netvisor Zrt. segítsége, azon belül is kiemelkedő *Máthé János, Schnell György, Szabó Balázs* és *Szemethy Tivadar* munkája és segítőkészsége. Emellett köszönet illeti a kutató és fejlesztő munkát végző illetve segítő munkatársakat is, név szerint *Do Van Tien, Jereb László, Horváth Ádám, Szalai László* és *Baráth Zsófia*.

IRODALOM

Boros J., Do, T.V., Horváth Á., Szalai L. (2010): Linux rendszerek teljesítmény-analízise. RET kutatási jelentés, NymE FMK INGA, 3. oldal

Jereb L., Do, T.V., Boros J., Horváth Á. (2011): Mérőszerver fejlesztése PVSR alá. RET kutatási jelentés, NymE FMK INGA, 3. oldal

Eckert J., Repp N., Schulte S., Berbner R., Steinmetz R. (2007): An approach for capacity planning of web service workflows. AMCIS 2007 Proceedings. Paper 80. [Online] <http://aisel.aisnet.org/amcis2007/80>

Menascé D.A., Almeida V.A.F. (2002): Capacity planning for Web services. Prentice Hall, 2. p.

Levelezési cím (*Corresponding author*):

Boros János
Nyugat-magyarországi Egyetem,
Faipari Mérnöki Kar, Informatikai és Gazdasági Intézet
9400 Sopron, Bajcsy-Zsilinszky u. 9.
University of West Hungary,
Faculty of Wood Sciences, Institute of Informatics and Economics
H-9400 Sopron, Bajcsy-Zsilinszky u. 9.
Tel.: 36-99-518-486, Fax: 36-99-518-367
e-mail: borosj@inf.nyme.hu