TELJESÍTMÉNYMÉRÉS FELHŐ ALAPÚ KÖRNYEZETBEN - AZURE CLOUD ANALÍZIS

Hartung István, hartung@iit.bme.hu
Dr. Goldschmidt Balázs, balage@iit.bme.hu
Budapesti Műszaki és Gazdaságtudományi Egyetem, Irányítástechnika és Informatika Tanszék
1117 Budapest, Magyar tudósok krt. 2

Tematika

A napjainkban egyre nagyobb teret nyerő felhő technológia lehetővé teszi, hogy a felhasználók ne a saját rendszerük erőforrásait, számítási kapacitását és háttértárát használják, hanem programokat és szolgáltatásokat telepítsenek távoli szerverekre. Egyre több cég szervezi ki szolgáltatásait különböző privát vagy publikus felhőkbe, hogy a központilag menedzselt és karbantartott szerverparkokban futó alkalmazások futtatása minél kevesebb költséggel járjon. Ugyanakkor rengeteg olyan szolgáltatás létezik, amelynél üzletileg kritikus a különböző vállalt a QoS (Quality of Service) értékek garantálása. Ezek azonban egy felhő alapú környezetben jelentősen ki vannak téve a szolgáltató által biztosított QoS értékeknek. Ezen felül az alkalmazások teljesítményére befolyással lehet rengeteg olyan –ritkán előforduló- azonban egyáltalán nem befolyásolható folyamat, amelyek szintén a felhő alapú környezetnek köszönhetőek. Kutatásunkban az Azure cloud szolgáltatását vizsgáltuk meg egy nagy terhelést kapó alkalmazás esetén, és azt vizsgáltuk, hogy milyen nem a rendes működéshez kapcsolódó, egyéb terheléseknek van kitéve az alkalmazást futtató virtuális gép automatikus teljesítménynaplózás vagy az alkalmazás újratelepítése esetén.

Bevezetés

A számítási felhő használatának számos oka lehet egy vállalatnál, de végső soron a költségmegtakarítás a fő célja bevezetésének.[1] A költségek csökkentését okozhatja, hogy cloud használatával hatékonyabbá válik a fejlesztés, a tesztelés, a bonyolult és hosszú telepítési ciklus lerövidül és hibamentessé válik. Szükségtelenné válik új szerverek vásárlása, konfigurálása és karbantartása, mivel gyorsabban és olcsóbban hozzá lehet jutni virtualizáció segítségével a szükséges gépekhez. A cég korábban meglevő infrastruktúráját sokkal hatékonyabban tudja kihasználni az által, hogy az erőforrásokat szétosztja, és a nem használt hardvert energiatakarékos állapotban tartja.

Ugyanakkor nagy problémát tud jelenteni egy felhőbe telepített alkalmazás esetén a megfelelő szolgáltatási szint (SLA) megállapodások betartása. Egyik nagy publikus felhő szolgáltató sem kínál több kilences rendelkezésre állást [2][3]. Ez egyrészt köszönhető a bonyolult, összetett architektúrájú rendszereknek, másrészt annak, hogy néhány globális jellegű konfigurációs hiba akár pillanatok alatt hatással lehet egy szolgáltatóhoz telepített összes alkalmazásra.

Jelen kutatás egy Windows Azure [4] felhőben futó valós alkalmazás tesztelését célozza meg. Célunk, hogy bemutassunk egy módszert felhőben futó alkalmazások

tesztelésére, valamint, hogy rámutassunk a különböző sajátosságokra és kihívásokra, amik egy ilyen környezetben bekövetkezhetnek.

A következő alfejezetekben olvasható egy rövid áttekintés a cloud technológiáról és az Azure felhőről, ezután következik a mérési környezet és metodika bemutatása, és az első körös mérési adatok elemzése, végül a további kutatási lehetőségek bemutatása.

Windows Azure

A számítási felhő alapú technológia infrastruktúrát (*Infrasturcture as a Service - IaaS*), platformot (*Platform as a Service - PaaS*) vagy szoftvert (*Software as a Service - SaaS*) nyújthat szolgáltatásként. Elmondható, hogy nincsen széles körben elfogadott definíció a cloud computing fogalmához, de általánosan elmondható a legtöbb számítási felhőről, hogy a szolgáltatást feliratkozás alapján lehet igénybe venni, úgy, hogy csak a valós erőforrás használatért kell fizetni (*pay-per-use*), amelynek mértéke rugalmasan változtatható, az erőforrások végtelen illúzióját keltve. Az önkiszolgáló interfészeken keresztül virtualizált hardver igényelhető, a valós fizikai konfiguráció a felhasználó elől mindig rejtve marad.

A Windows Azure Platform a Microsoft felhő szolgáltatása, amely mind IaaS, mind PaaS szolgáltatásokat nyújt. Infrastruktúra alapú szolgáltatásként az Azure lehetőséget biztosít perzisztens virtuális gépek futtatására, amelyeken mind Windows, mind néhány Linux alapú operációs rendszer (közös követelmény, hogy futni tudjon rajta az Azure saját ügynökprogramja, amely biztosítja a virtuális gép monitorozását).

A felhő szolgáltató emellett biztosít számos platform alapú szolgáltatást. Lehetőség van előre csomagolt, akár több külön gépen futó szerepeket tartalmazó programcsomagot feltelepíteni a felhőbe, ahol az egy vagy több újonnan indított virtuális gépre lesz telepítve, amelyek karbantartását, frissítését, menedzsmentjét a Windows Azure végzi el automatikusan. Ennek ellenére lehetőség van távoli asztallal belépni a gépekre, és azokról tetszőleges adatokat nyerni, azonban az elvégzett változtatások bármikor visszavonódhatnak, mivel az automatikus menedzsment operációknak köszönhetően bármikor új szerver indulhat a feltelepített programcsomag kiszolgálására.

További platform szintű szolgáltatásokat jelentenek a különböző perzisztens adattárolást lehetővé tevő komponensek. Ezek közül az első az Azure SQL, amely egy enyhén csökkentett képességű [5] Microsoft SQL Server implementáció, amely lehetőséget nyújt akár 100 GB adat tárolására, és biztosít ezen adatok eléréséhez tranzakció kezelést és egy teljesen általános SQL interfészt (TDS protokoll segítségével). Ennél nagyobb adatbázis méret esetén lehetőség van az adatok szétosztására több adatbázis szerver példány között (Azure SQL Federation).

További adattárolási lehetőséget biztosítanak az Azure Storage megoldásai. Ezek közül a legfontosabbak a nagyméretű bináris fájlok perzisztens tárolását lehetővé tevő Blob Storage (Binary Large Object Storage), a skálázható, párhuzamosan is elérhető NoSQL táblákat biztosító Azure Tables adattár, valamint a szerver példányok közti aszinkron üzenettovábbítást lehetővé tevő Azure Queue. Mindegyik szolgáltatás biztosít az adatokról biztonsági másolatot, akár geográfiailag másik adatközpontba is, így biztosítván, hogy ne fordulhasson elő adatvesztés.

Ezek mellett a Windows Azure cloud biztosít még számos másik szolgáltatást (weboldal hosztolás, Enterprise Service Bus szolgáltatás, virtuális hálózatok kiépítése,

adatszinkronizáció adatbázisok között, mobil szolgáltatások, push notification, média streaming, Active Directory), azonban ezeket jelen kutatás nem érinti.

Mérési környezet

A kutatás során a vizsgált alkalmazás egy valós környezetben is működő, éles alkalmazás volt, amely Windows Azure cloudban futott, ahol REST-es API hívásokon keresztül volt megszólítható, HTTPS protokoll használatával, kliens tanúsítvány alapú authentikációval.

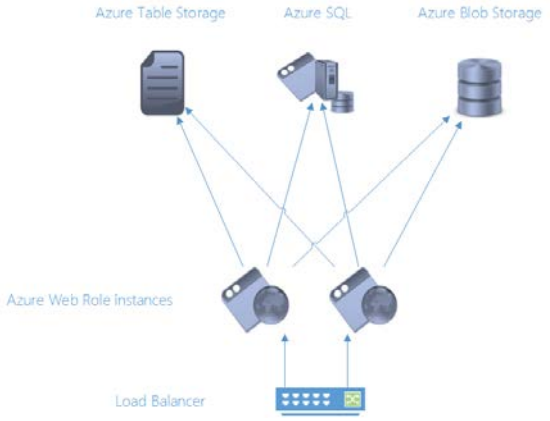
Az alkalmazás két *Azure Small Instance* virtuális gépen fut, amelyek közös címen érhetőek el, egy terheléselosztó mögött. Külön-külön kívülről nem lehet megkülönböztetni a gépeket, valamint nem befolyásolható az sem, hogy a terheléselosztó melyik példányhoz irányítja a beérkező kérést.

Az egyes virtuális gépek egy közös Azure SQL adatbázis szolgáltatást érnek el, ennek tárolt eljárásait szólítják meg. Az adatbázist az Azure szolgáltatja, annak naplójához semmilyen módon nem lehet kívülről hozzáférni valamint ez is több valós fizikai gép példányon fut, amelyek előtt egy terheléselosztó található.

A webalkalmazások az Azure cloud sajátosságainak köszönhetően semmit sem tárolhatnak perzisztens módon lokálisan. Tartós adatokat az adatbázisban és egy Azure Blob tárolóban tárol az alkalmazás. Ehhez mind írási mind olvasási műveleteket végez. A perzisztens teljesítménynaplót és alkalmazásnaplót egy NoSQL adatbázisban tárolja, amelyet szintén az Azure szolgáltat (Azure Tables).

Az alkalmazás ezen kívül igénybe vesz egy külső szolgáltató által biztosított SMTP szolgáltatást, amelyen keresztül bizonyos események hatására email üzenetet küld ki a felhasználók számára. Ezt a szolgáltatást azonban SMTP helyett REST alapú interfészen keresztül veszi igénybe az Azure korlátozásai miatt.

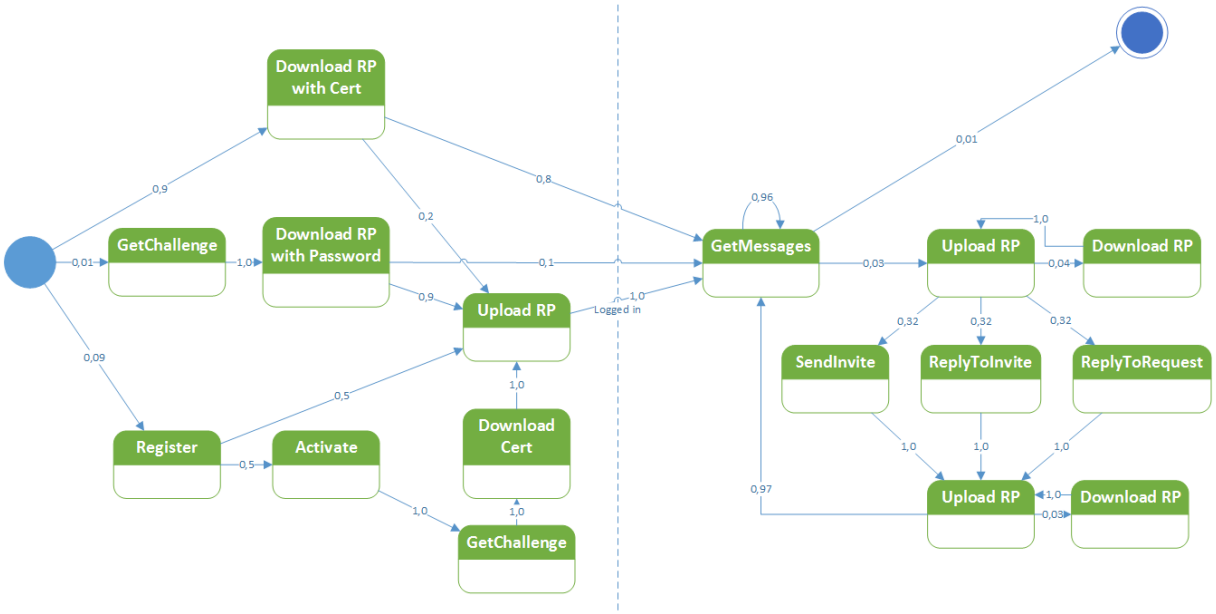
A webalkalmazás architektúrája az 1. ábráról olvasható le.



1. ábra A tesztelt alkalmazás architektúrája

A szolgáltatás közel húsz különböző operációt biztosít, amelyeket azonban valós esetben a kliensek közel sem azonos valószínűséggel hívnak. Az egyes hívások előfordulási aránya a korábbi használati statisztikákból adott volt a mérés kezdetén.

Emellett nem elhanyagolható az sem, hogy az egyes hívásoknak mi a sorrendje egy adott kliens tekintetében, hiszen ugyan állapotmentes a szerver (nincs session információ). Azonban a perzisztens módon tárolt adatoknak köszönhetően nem elhanyagolható, hogy az egyes hívások milyen sorrendben futnak le egy-egy kliens tekintetében, hiszen ez hatással lehet mind a szerver belső teljesítményére (különböző automatikus cache-eléseknek köszönhetően), illetve a visszaadott adatokra mind méret, mind tartalom szempontjából.



2. ábra A kliens CBMS modellje

Jelen esetben az egyes állapotok a kliens által hívott szerver oldali metódusokat jelölik. Ez a modell látható a 2. ábrán. Logikailag két fázisra osztható a rendszer. A kezdeti fázisban történik meg a bejelentkezés, majd utána a rendes működés látható. Ezen hívások nagy része automatikusan történik, nagyon kevés átmenet esetén lehetséges felhasználói interakció.

Ami a modellből nem látszódik az ábrázolás miatt, hogy belépés után az esetek túlnyomó többségében a *GetMessages()* függvény fog hívódni. Ennek lefutása után azonban a végrehajtó szál pihen, 15 másodpercig nem szólítja meg a felhőben futó alkalmazást az eredeti kliens szoftver. Ezt a megkötést vettük figyelembe a későbbiekben a tesztkliens megírásakor.

Terheléses teszteléshez az eredeti kliens nem használható jól, mivel felhasználói interakciót igényel a legtöbb művelet. amely végrehajtható vele, valamint az egyes hívások hosszának mérése is szükséges, hiszen ez szolgál visszajelzést a szolgáltatásminőséggel kapcsolatosan. Teszteléshez így egy saját készítésű kliens készült, amely képes szimulálni egy felhasználó viselkedését a szerver szempontjából, valamint értelmes tartalmú üzeneteket tud küldeni, betartván a korábban meghatározott állapotdiagram szerinti működést. A kliens emellett naplózza az egyes kérések átfutási idejét (round trip time) és az adatokat időbélyeggel együtt menti egy naplófájlba.

Az első mérés a felhőben futó alkalmazást lokálisan futtatott tesztklienssel, egyszerre pár száz felhasználót szimulálva készült. Ennek során kiderült, hogy a szűk keresztmetszetet a tesztrendszerben nem a felhőben futó alkalmazás, hanem a tesztklienst futtató gép és annak hálózati kapcsolata jelentette.

Ennek köszönhetően maga a tesztalkalmazás is felkerült az Azure felhőbe, hogy a felhő belső hálózatát használva a hálózat már ne jelenthessen szűk keresztmetszetet, valamint így biztosítván, hogy a tesztelés során tetszőleges számú klienst lehessen indítani. A Windows Azure adattárolási lehetőségeit kihasználván a tesztkliensek így nem lokálisan naplózták a korábban

meghatározott adatokat, hanem ezeket Azure Tablesben tárolták el.

A második mérés során nyolc szerver példányról futott a tesztkliens, egyszerre párhuzamosan közel kétezer felhasználót szimulálva. Ennek a mérésnek az eredményei olvashatóak a továbbiakban.

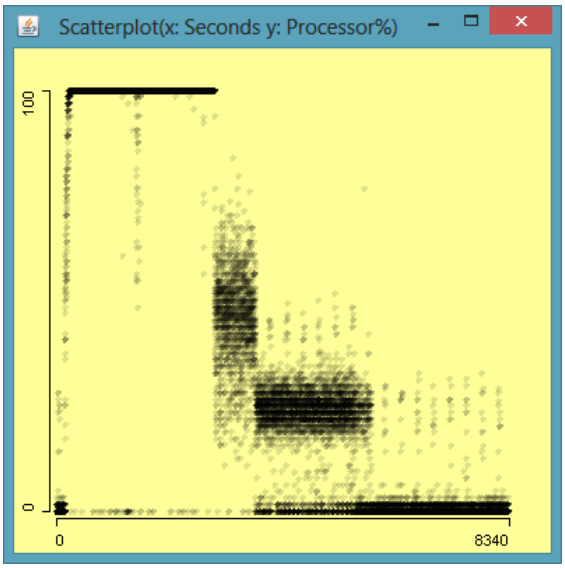
A mérés idejére a szerver alkalmazásban be lett konfigurálva az automatikus teljesítménynaplózás, amely több virtuális gép szintű metrikát naplózott, másodperc finomságú felbontással. Az adatok az Azure automatikus teljesítménynaplózó megoldásával (Diagnostics) kerültek eltárolásra Azure Tablesben.

Eredmények

A mérés eredményét jelentő teljesítményadatokat egy egyedi kliens segítségével lehetett letölteni az Azure Tables adatbázisból, majd azt vizualizálható formára kellett hozni. Az eredmények megjelenítéséhez a Mondrian szoftvert használtuk. [6]

Az eredményekből kiderül, hogy sikerült az alkalmazást futtató mindkét szervert megterhelni CPU használt szempontjából (3. ábra). Azonban látszik az eredményekből, hogy a memóriahasználat egyáltalán nem jelentős az egyes gépeken, így elmondható, hogy a mért alkalmazás nem végez memória intenzív műveleteket.

A processzor terhelés vizsgálata során feltűnő jelenség, hogy több mint fél órával azután, hogy a terhelés abbamaradt még mindig intenzív CPU használt volt mindkét virtuális gépen, holott semmilyen beérkező kérés nem jött már ekkor a rendszerbe.

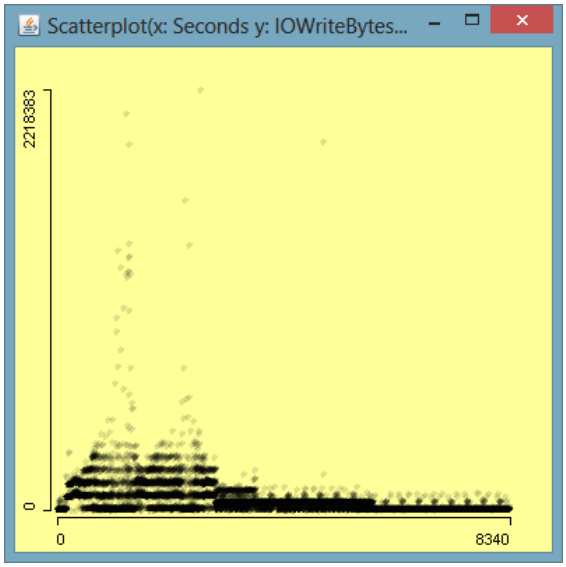


3. ábra Az egyes virtuális gépek CPU használata az idő függvényében

A jelenség magyarázatához az IO műveletek vizsgálata segített hozzá. Az eredeti szerver alkalmazás semmilyen módon nem használja a virtuális gép lemezét, nem ír vagy olvas semmilyen adatot onnan. Azonban megvizsgálva a teljesítménynapló által rögzített értékeket (4. ábra) több érdekesség is látszódik. Pontosan öt percenként van egy intenzív IO írási művelet, amely miatt kiugrik a rendszer lemezhasználata (ezzel párban van egy hasonló mértékű lemezolvasási művelet is), valamint, hogy egészen addig, míg nem csillapodik közel konstans nullára a CPU használat, van folyamatos lemezhasználat is.

A mért jellemzők megmutatják, hogy az Azure platform hogyan menedzseli automatikusan a naplózási információkat. Az öt percenkénti kiugró érték annak köszönhető, hogy a virtuális gépek ilyen időközönként viszik át az addig memóriában, illetve lokálisan diszken tárolt erőforrás használati napló adatait a perzisztens Azure Tables tárolóba. A konstans jellegű magasabb szintű IO használat pedig a beérkező kérések naplózásának köszönhető. A rendszer automatikusan naplózza az egyes beérkező kéréseket, amelynek perzisztálását szintén az Azure platform teszi meg

automatikusan. Ennek köszönhető, hogy az alaposan leterhelt rendszer, amely sikeresen kiszolgálta az összes bejövő kérést, a kérések befejezte utáni fél órában végzi el az alapvetően kisebb prioritású műveletet, amelynek során a lokálisan rögzített információk átkerülnek Azure Tablesbe.



4. ábra Az egyes virtuális gépek IO írás/másodperc értéke az idő függvényében

További kutatási irány

Jelen kutatásban célunk, hogy feltérképezzük az Azure szolgáltatás határait nagy terhelés esetén, és ezáltal további információt nyerjünk belső működésével kapcsolatban, valamint kidolgozzunk egy általánosan használható metodikát felhő környezetben való teszteléshez.

Érdekes lehet többet megtudni az Azure egyes adattárolási PaaS szolgáltatásainak határairól, amelyek akár olyan tudományos számítások során is hasznosak tudnak lenni, ahol a felhő által nyújtott számítási és adattárolási kapacitás nélkül szuperszámítógépre lenne szükség.

Nem akadémiai környezetben a kutatás jelentősége lehet, hogy megmutassa, hogy milyen jellegű alkalmazások esetén mely erőforrásokat milyen célból érdemes igénybe venni egy publikus felhő szolgáltatónál, hogy az alkalmazás megfelelő szolgáltatásminőséget nyújtva ki tudjon szolgálni egyszerre tetszőleges mennyiségű klienst.

Összefoglalás

Jelen kutatás során megvizsgáltuk az Azure platformot egy valós életbeli alkalmazás teljesítménymérése kapcsán. A kutatás során bemutattunk egy technikát, ahogy valós életbeli terhelés generálható egy alkalmazás teljesítményméréséhez felhő alapú környezetben. A vizsgálat során megmutattuk, hogy hogyan működik a Windows Azure felhőben az automatikus naplózás szolgáltatás. Célunk a rendszer további vizsgálata, az egyes PaaS szolgáltatások vizsgálata nagy terhelés esetén.

Köszönetet mondunk mindazoknak (BME VIK hallgatóinak, egyetemi kutató kollégáknak), akik munkájukkal, tanácsukkal és szakértelmükkel segítették jelen kutatást.

A munka szakmai tartalma kapcsolódik a "Új tehetséggondozó programok és kutatások a Műegyetem tudományos műhelyeiben" c. projekt szakmai célkitűzéseinek megvalósításához. A projekt megvalósítását a TÁMOP-4.2.2.B-10/1--2010-0009 program támogatja.

Hivatkozások

[1] J. Staten, S. Yates, F. E. Gillett, W. Saleh: Is Cloud Computing Ready For The Enterprise? (2008, *Forrester Research*)

[2] Azure cloud SLA (Service Level Agreement): http://www.windowsazure.com/en-us/support/legal/sla/

[3] Amazon EC2 SLA: http://aws.amazon.com/ec2-sla/

[4] Azure cloud: http://www.windowsazure.com

[5] Azure SQL és SQL Server különbségek http://social.technet.microsoft.com/wiki/contents/articles/996.compare-sql-server-with-windows-azure-sql-database.aspx

[6] Mondrian http://www.theusrus.de/Mondrian/