

Általános információk, a diplomaterv szerkezete

A diplomaterv szerkezete a BME Villamosmérnöki és Informatikai Karán:

1. Diplomaterv feladatkiírás
2. Címoldal
3. Tartalomjegyzék
4. A diplomatervező nyilatkozata az önálló munkáról és az elektronikus adatok kezeléséről
5. Tartalmi összefoglaló magyarul és angolul
6. Bevezetés: a feladat értelmezése, a tervezés célja, a feladat indokoltsága, a diplomaterv felépítésének rövid összefoglalása
7. A feladatkiírás pontosítása és részletes elemzése
8. Előzmények (irodalomkutatás, hasonló alkotások), az ezekből levonható következtetések
9. A tervezés részletes leírása, a döntési lehetőségek értékelése és a választott megoldások indoklása
10. A megtervezett műszaki alkotás értékelése, kritikai elemzése, továbbfejlesztési lehetőségek
11. Esetleges köszönetnyilvánítások
12. Részletes és pontos irodalomjegyzék
13. Függelék(ek)

Felhasználható a következő oldaltól kezdődő L^AT_EX-diplomaterv sablon dokumentum tartalma.

A diplomaterv szabványos méretű A4-es lapokra kerüljön. Az oldalak tükörmargóval készüljenek (min-denhol 2,5 cm, baloldalon 1 cm-es kötéssel). Az alapértelmezett betűkészlet a 12 pontos Times New Roman, másfeles sorközzel.

Minden oldalon – az első négy szerkezeti elem kivételével – szerepelnie kell az oldalszámnak.

A fejezeteket decimális beosztással kell ellátni. Az ábrákat a megfelelő helyre be kell illeszteni, fejezetenként decimális számmal és kifejező címmel kell ellátni. A fejezeteket decimális alaosztással számozzuk, maximálisan 3 alaosztás mélységben (pl. 2.3.4.1.). Az ábrákat, táblázatokat és képleteket célszerű fejezetenként külön számozni (pl. 2.4. ábra, 4.2. táblázat. vagy képletnél (3.2)). A fejezetcímeket igazítsuk balra, a normál szövegnél viszont használjunk sorkiegyenlítést. Az ábrákat, táblázatokat és a hozzájuk tartozó címet igazítsuk középre. A cím a jelölt rész alatt helyezkedjen el.

A képeket lehetőleg rajzoló programmal készítsék el, az egyenleteket egyenlet-szerkesztő segítségével írják le (A L^AT_EX ehhez kézenfekvő megoldásokat nyújt).

Az irodalomjegyzék szövegek közötti hivatkozása történhet a Harvard-rendszerben (a szerző és az évszám megadásával) vagy sorszámozva. A teljes lista névsor szerinti sorrendben a szöveg végén szerepeljen (sorszámozott irodalmi hivatkozások esetén hivatkozási sorrendben). A szakirodalmi források címét azonban mindig az eredeti nyelven kell megadni, esetleg zárójelben a fordítással. A listában szereplő valamennyi publikációra hivatkozni kell a szövegben (a L^AT_EX-sablon a Bib_TE_X segítségével mindezt automatikusan kezeli). Minden publikáció a szerzők után a következő adatok szerepelnek: folyóirat cikkeknel a pontos cím, a folyóirat címe, évfolyam, szám, oldalszám tól-ig. A folyóirat címeket csak akkor rövidítsük, ha azok nagyon közismertek vagy nagyon hosszúak. Internet hivatkozások megadásakor fontos, hogy az elérési út előtt megadjuk az oldal tulajdonosát és tartalmát (mivel a link egy idő után akár elérhetetlenné is válhat), valamint az elérés időpontját.

Fontos:

- A szakdolgozat készítő/diplomatervező nyilatkozata (a jelen sablonban szereplő szövegtartalommal) kötelező előírás Karunkon ennek hiányában a szakdolgozat/diplomaterv nem bírálható és nem védhető!
- Mind a dolgozat, mind a melléklet maximálisan 15 MB méretű lehet!

Jó munkát, sikeres szakdolgozat készítést, ill. diplomatervezést kívánunk!

Feladatkiírás

A feladatkiírást a tanszéki adminisztrációban lehet átvenni, és a leadott munkába eredeti, tanszéki pecséttel ellátott és a tanszékvezető által aláírt lapot kell belefűzni (ezen oldal *helyett*, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatba már nem kell beleszerkeszteni ezt a feladatkiírást.



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Méréstechnika és Információs Rendszerek Tanszék

Hibatűrő Rendszerek Kutatócsoport

Felhő alkalmazások automatikus skálázási algoritmusainak benchmarkolása

SZAKDOLGOZAT

Készítette

Prontvai Zsolt

Konzulens

Kocsis Imre

2015. november 24.

Tartalomjegyzék

HALLGATÓI NYILATKOZAT

Alulírott *Prontvai Zsolt*, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2015. november 24.

Prontvai Zsolt
hallgató

Kivonat

Abstract

1. fejezet

Cloud alapok

1.1. Mi a cloud computing

1.2. Cloud rendszerek szolgáltatás szerinti csoportosítása

1.2.1. SaaS

1.2.2. PaaS

1.2.3. IaaS

1.3. Cloud rendszerek szervezés szerinti csoportosítása

1.3.1. Public cloud

1.3.2. Private cloud

1.3.3. Hibrid cloud

2. fejezet

IaaS cloudok

2.1. Amazon Web Services

2.2. Windows Azure

2.3. Google Compute Engine

2.4. Rackspace Open Cloud

2.5. IBM SmartCloud Enterprise

2.6. HP Enterprise Converged Infrastructure

3. fejezet

Skálázási módszerek

3.1. Miért szükséges

3.2. Módszerek

3.2.1. Maximumra tervezés

3.2.2. Átlagra tervezés

3.2.3. Dinamikus erőforrás allokáció

3.3. Főbb típusok

3.3.1. Scaling up

3.3.2. Scaling out

4. fejezet

Automatikus skálázás algoritmusok

Skálázó algoritmusokat általánosságban két csoportba sorolhatjuk. Egy algoritmus lehet reaktív, mely a rendszer egy valamilyen állapotára reagál, például érzékelik, hogy kevés az erőforrás, akkor allokal újabbakat. A proaktív algoritmusok ezzel szemben előre próbálják jelezni a rendszer elkövetkezendő állapotát, hogy az erőforrások rendelkezésre álljon amikor később szükség lesz rájuk.

4.1. Ütemterv alapú

Számításba veszi a forgalom napszakok szerinti változását. Az az erőforrás allokaláció mauálisan van beprogramozva a napszaktól függően. Például a délutáni időszakban nagyobb forgalom várható ezért ilyenkor több erőforrásra van szükség, míg az éjszaka folyamán kisebb a forgalom ezért ilyenkor kevesebb erőforrásra van szükség, mint az átlag. Nem tud reagálni a forgalomba bekövetkező váratlan változásokra.

4.2. Szabály alapú

A legtöbb szolgáltató ezt használja. Ehhez az egyszerű megközelítéshez általában két szabályt kell megfogalmazni, hogy mikor skálázzunk fel, illetve mikor skálázzunk le. A felhasználónak valamilyen metrika szerinti feltételt kell megfogalmaznia. Például, ha az átlagos processzor kihasználtság nagyobb mint 70%. Ha a feltétel teljesült, akkor automatikusan triggerelődik az előre definiált skálázási akció. Például elindít egy újabb virtuális szerver. A szabály alapú automatikus skálázó algoritmusokat, reaktív algoritmusok közé sorolhatjuk.

4.3. Idősor elemzés

A idősor elemzésnél bizonyos metrikákat, például az átlagos processzor kihasználtságot, vagy a beérkező terhelést megadott időközönként mintavételezzük. Az eredmény egy idősor lesz ami tartalmazza az elmúlt n megfigyelés sorozatát. A automatikus skálázás

problémája két részre bontható az elkövetkezendő metrika előrejelzése, illetve az ebből következő döntéshozás. Az idősor elemzés csak az első felére tudjuk alkalmazni, a másodikra használatunk például egyszerű szabálya alapú skálázást. A idősor elemzésnek két célja van, előre jelezni a jövő beli értéket az elmúlt megfigyelések alapján, illetve mintákat felismerni, melyből extrapolálható a jövőbeli érték.

Módszerek

- Átlag
- Mozgó átlag
- Súlyzott mozgó átlag
- Exponenciális simítás
- Autó-regresszió
- Autó-regressziós mozgó átlag
- Gépi tanulás

Minták

- Minta felismerés
- Jelfeldolgozó technikákat
- Auto-korreláció

4.4. Szabályozás elmélet

A szabályozó rendszereknek három fajtája van open loop, feedback and feed-forward

4.4.1. Nyílt

Csak a rendszer jelenlegi állapotát és a rendszerről alkotott modelt. Nem használ visszacsatolást, hogy a rendszer elérte a kívánt végeredmént

4.4.2. Visszacsatolt

Figyelik a rendszer kimenetét, és korrigálják az eltérést a kívánt céltól. Fixed gain controllers
Adaptive control Reconfiguring control Model predictive control

4.4.3. Előreccsatolt

Megjósolják a rendszer viselkedését, a róla alkotott modell alapján, és megelőzik a hibát mielőtt bekövetkezne.

4.5. Megerősítéses tanulás

A megerősítéses tanulás egy automatikus döntéshozó algoritmus, amelyet alkalmazhatunk automatikus skálázásra. A döntéshozó (ágens), a tapasztalataiból (próbálgatás) tanulja meg hogy melyik a legjobb művelet amit végre kell hajtania a környezete bármely állapotában, az

által, hogy maximalizálni próbálja az érte kapott jutalmat. A mi esetünkben az autmatikus skálázó az ágens, aki kapcsolatban van a skálázandó alkalmazással (környezet). Dönteni fog hogy adjon, illetve elvegyen erőforrásokat az alkalmazástól (művelet), a jelenlegi terheléstől, teljesítménytől és egyéb változóktól (állapot) függően, úgy hogy minimalizálja az alkalmazás válaszidejét vagy egyébbb változóit (jutalom).

4.6. Sorbanállás-elmélet

A sorbanállás-elmélet különböző folyamatok eseményeivel kapcsolatos várakozási sorokat, sorbanállási időket a kiszolgálásra, és ezek összefüggéseit tárgyalja az alkalmazott matematikai eszközeivel. A sorbanállási elméletben becslési modellt alkotnak a sorbanállás hosszáról és időtartamáról, és a kiszolgálás sikerességéről. Az egyszerű csomóponti sorbanállásokat a Kendall-féle jelöléssel jellemezznek $A/B/C/K/N/D$ formában, ahol A írja le a beérkezési időközt, B a munka nagyságát (kiszolgálási idő), és C a kiszolgálók számát, K a rendszer kapacitását vagy a sor hosszát, N az igényforrás számossága, D a kiszolgálás elve vagy a fontossági sorrend. A mi esetünket megfeleltethetjük egy egyszerű csomóponti sorbanállásnak, úgy hogy egy sor megy a terhelés elosztóhoz, ami elosztja a kéréseket a virtuális szerverek felé.

5. fejezet

Skálázási algoritmusok minőségmodelje

5.1. Latency

5.2. Throughput

5.3. Efficiency

5.4. Üzemeltetési költség

5.5. SLA megsértés

6. fejezet

Terhelési minták

6.1. Tipusok

6.1.1. Stabil

6.1.2. Növekvő

6.1.3. Ciklus

6.1.4. Ki és be

6.2. Slashdot effect

7. fejezet

Keretrendszer

7.1. AWS

7.1.1. Instance típusok

7.1.2. IAM

7.1.3. AMI

7.1.4. ELB

7.1.5. Autoscaling group

7.2. Wordpress

7.3. JMeter

7.4. Ruby

7.5. Graphite

7.5.1. Collectd

7.5.2. Statsd

8. fejezet

Mérési eredmények

8.1. Google Trace

8.2. Mozgó átlag

8.3. exponenciális simítás

8.4. ARMA

9. fejezet

Konklúzió

10. fejezet

Összefoglaló

A diplomaterv összefoglaló fejezete.

Köszönetnyilvánítás

A köszönetnyilvánítás nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Táblázatok jegyzéke

Ábrák jegyzéke

Irodalomjegyzék

A. függelék

Függelék