5. gyakorlat – Teljesítménymodellezés – Megoldások

1. Diszk teljesítménye

Egy diszk 50 kérést szolgál ki másodpercenként. Minden kérés kiszolgálása 0,005 másodpercet vesz igénybe. A rendszerben nincs átlapolódás.

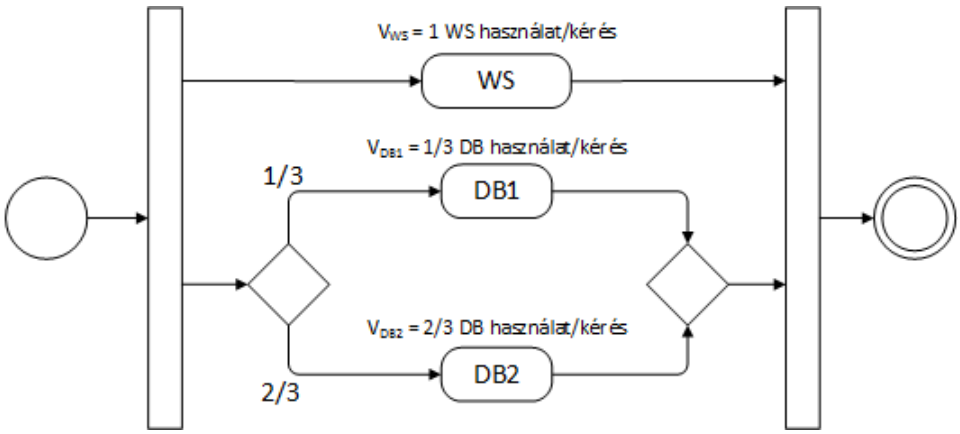
a) Mekkora a kihasználtság?
Megoldás
Az erőforrás kihasználtsága $U = X \cdot T$, ahol X az átlagos átbocsájtás és T az átlagos kiszolgálási idő. Tehát $U = 0{,}25$, így 25%-os a kihasználtság.
A feladat józan ésszel is megoldható: a diszknek másodpercenként 50 kérés $\cdot$ 0,005 $\frac{\text{s}}{\text{kérés}}$-t kell dolgoznia. Ha másodpercenként 0,25 másodpercet dolgozik, akkor 25% a kihasználtsága.

b) Mekkora a maximálisan kiszolgálható terhelés (érkezési ráta)?
Megoldás
Maximális terhelés mellett a kihasználtság $U = 1$. Ekkor $X_{\max} = \frac{U}{T} = 200\ \frac{\text{kérés}}{\text{s}}$. Vagyis a szabály egyetlen, átlapolódásmentes feldolgozó egységre: $X_{\max} = \frac{1}{T} = \frac{1}{0{,}005\ \text{s}} = 200\ \frac{\text{kérés}}{\text{s}}$.

2. Kétrétegű architektúra

Adott egy webszerver (WS) és két fürtözött adatbázisszerver (DB1, DB2). A két adatbázis szerver közt súlyozott round robin terheléselosztás alapján választunk, 1:2 arányban. Minden felhasználói kérés kiszolgálása során mindkét fajta erőforrást használjuk. A csúcsidőszakban 30 percig monitorozzuk a rendszert, ezalatt 9000 kérést szolgál ki. A szerveken mért foglaltsági idők: WS – 1350 s CPU idő; DB1 – 810 s, DB2 – 1320 s diszk IO idő.

a) Készítsünk folyamatmodellt a kérések feldolgozásáról a szöveg alapján!
Megoldás



Mivel a feladatban nem volt egyéb megkötés, azt feltételeztük, hogy a kérések kiszolgálása a különböző erőforrásokon párhuzamosan történik. Ehelyett a modell lehetne szekvenciális is (az átbocsátás szempontjából nincs különbség, *de a végrehajtási időben igen*!), viszont az előbbi általánosabb, hiszen a WS használata átlapolódhat az adatbázis használatával. A valóságban persze a WS az adatbázishívás előtt és után is dolgozik, sőt, időnként még közben is. A mostani modell azt fejezi ki, hogy – pontos információ híján – ezeket a szakaszokat aggregáljuk és elfelejtjük, hogy milyen sorrendben futottak (absztrakció!).

b) Mekkora az egyes szerverek jelenlegi átbocsátása?
Megoldás
Emlékeztető: A vizitációs számmal (többek között) a rendszer és a komponensek átbocsátása és átbocsátóképessége között tudunk váltani. Ha átbocsátással dolgozunk, akkor rendszerint a rendszer átbocsátásából számítjuk a komponensek átbocsátását – ilyenkor a vizitációs számmal szorozni kell, hiszen minden rendszerbe belépő tokent átlagosan annyiszor kell feldolgoznia a komponenseknek, mint amennyi a vizitációs szám. Ha átbocsátóképességet szeretnénk számolni, akkor rendszerint a komponensek (egyszerűen számítható) átbocsátóképességéből kiindulva határozzuk meg a rendszer átbocsátóképességét – ilyenkor a vizitációs számmal osztani kell, hiszen

ha minden belépő tokent annyiszor kell feldolgoznia a rendszernek, mint amennyi a vizitációs szám, akkor annyival kevesebb token érkezhet a rendszerbe túltelítődés nélkül. Ne feledjük, hogy (többek között a szűk keresztmetszetek miatt) ebben az irányban nem elegendő a vizitációs számmal számolni, gyakran szükség van a számított értékeken végzett egyéb számításokra (pl. minimumképzésre)

Számoljunk először a rendszerre, aztán az erőforrásokra! A feldolgozott kérések száma $C = 9000$ („Count"), a mérés ideje $T_m = 30$ min.

- $X_{\text{rendszer}} = \frac{C}{T_m} = \frac{9000 \text{ kérés}}{30 \text{ min}} = \frac{9000}{1800} \frac{\text{kérés}}{\text{s}} = 5 \frac{\text{kérés}}{\text{s}}$
- $X_{\text{WS}} = X_{\text{rendszer}} \cdot v_{\text{WS}} = 5 \frac{\text{kérés}}{\text{s}} \cdot 1 = 5 \frac{\text{kérés}}{\text{s}}$
- $X_{\text{DB1}} = X_{\text{rendszer}} \cdot v_{\text{DB1}} = 5 \frac{\text{kérés}}{\text{s}} \cdot \frac{1}{3} = 1{,}666 \frac{\text{kérés}}{\text{s}}$
- $X_{\text{DB2}} = X_{\text{rendszer}} \cdot v_{\text{DB2}} = 5 \frac{\text{kérés}}{\text{s}} \cdot \frac{2}{3} = 3{,}333 \frac{\text{kérés}}{\text{s}}$

c) Mennyi időt töltenek egy-egy kérés kiszolgálásával a szerverek?

Megoldás

Az egyes erőforrásokra (B a mért foglaltsági idő, „Busy time", az egyes szerverek pedig $C \cdot v_i$ kérést dolgoznak fel):

- $T_{\text{WS}} = \frac{B_{\text{WS}}}{C \cdot v_{\text{WS}}} = \frac{1350 \text{ s}}{9000 \text{ kérés}} = 0{,}15 \frac{\text{s}}{\text{kérés}}$
- $T_{\text{DB1}} = \frac{B_{\text{DB1}}}{C \cdot v_{\text{DB1}}} = \frac{810 \text{ s}}{3000 \text{ kérés}} = 0{,}27 \frac{\text{s}}{\text{kérés}}$
- $T_{\text{DB2}} = \frac{B_{\text{DB2}}}{C \cdot v_{\text{DB2}}} = \frac{1320 \text{ s}}{6000 \text{ kérés}} = 0{,}22 \frac{\text{s}}{\text{kérés}}$

d) Mekkora a rendszer maximális áteresztőképessége?

Megoldás

A rendszer maximális átbocsátóképessége az a legnagyobb átbocsátás, amivel egyik komponensbe sem érkezik több kérés, mint annak átbocsátóképessége. Ennek megfelelően pl. a DB1 ágra

$$X_{\text{rendszer}} \cdot v_{\text{DB1}} \leq X_{\text{DB1}}^{\max} \Rightarrow X_{\text{rendszer}} \leq \frac{1}{v_{\text{DB1}}} \cdot X_{\text{DB1}}^{\max}.$$

Ugyanígy DB2-re és WS-re:

$$X_{\text{rendszer}} \leq \frac{1}{v_{\text{DB2}}} \cdot X_{\text{DB2}}^{\max}$$

$$X_{\text{rendszer}} \leq \frac{1}{v_{\text{WS}}} \cdot X_{\text{WS}}^{\max} = X_{\text{WS}}^{\max}. \tag{1}$$

Mivel DB1 és DB2 *kötött arányú választás* (hosszú távon gyakorlatilag olyan, mintha minden „munkát" 1:2 arányban szétbontanánk és továbbküldenénk, tehát ilyen szempontból a fork-join és a szabad választás[1] közé tehető), ezért a számított értékek minimuma érkezhet meg a decision csomóponthoz túltelítés nélkül:

$$X_{\text{rendszer}} \leq \min\left(\frac{1}{v_{\text{DB1}}} \cdot X_{\text{DB1}}^{\max}, \frac{1}{v_{\text{DB2}}} \cdot X_{\text{DB2}}^{\max}\right). \tag{2}$$

A fork mindig mindkét irányba továbbküldi a kérést, és mindkét irányba a „teljes munkát" továbbítja, tehát az elágazásra számított érték és a WS-re számított érték közül a kisebb lehet a rendszer átbocsátóképessége. Ezalapján az 1 és a 2 egyenlőtlenségekből a maximális átbocsátás, vagyis az átbocsátképesség képlete:

$$X_{\text{rendszer}}^{\max} = \min\left(X_{\text{WS}}^{max}, \frac{1}{v_{\text{DB1}}} X_{\text{DB1}}^{max}, \frac{1}{v_{\text{DB2}}} X_{\text{DB2}}^{max}\right)$$

A feladat megoldásához tehát a komponensek átbocsátóképességeit kell kiszámolnunk:

- $X_{\text{WS}}^{\max} = \frac{1}{T_{\text{WS}}} = \frac{1}{0{,}15 \frac{\text{s}}{\text{kérés}}} = 6{,}666 \frac{\text{kérés}}{\text{s}}$
- $X_{\text{DB1}}^{\max} = \frac{1}{T_{\text{DB1}}} = \frac{1}{0{,}27 \frac{\text{s}}{\text{kérés}}} = 3{,}704 \frac{\text{kérés}}{\text{s}}$
- $X_{\text{DB2}}^{\max} = \frac{1}{T_{\text{DB2}}} = \frac{1}{0{,}22 \frac{\text{s}}{\text{kérés}}} = 4{,}545 \frac{\text{kérés}}{\text{s}}$

A rendszer maximális átbocsátóképessége ezekből:

[1] A szabad választású döntés akármelyik irányba továbbküldheti a kérést, tehát ha az egyik ág telítésben van, nyugodtan választhatja a másikat (a kötött arányú nem). Emiatt szabad választásnál az átbocsátóképességek összeadódnak.

$X^{\max}_{\text{rendszer}} = \min\left(6{,}666\ \frac{\text{kérés}}{\text{s}}, 3 \cdot 3{,}704\ \frac{\text{kérés}}{\text{s}}, \frac{3}{2} \cdot 4{,}545\ \frac{\text{kérés}}{\text{s}}\right) =$
$\min\left(6{,}666\ \frac{\text{kérés}}{\text{s}}, 11{,}112\ \frac{\text{kérés}}{\text{s}}, 6{,}818\ \frac{\text{kérés}}{\text{s}}\right) = X^{\max}_{\text{WS}} = 6{,}666\ \frac{\text{kérés}}{\text{s}}$.
Érdemes megfigyelni, hogy a minimum a WS-en esett, de a DB2-höz tartozó érték ($6{,}818\ \frac{\text{kérés}}{\text{s}}$) szintén nagyon közel van. A szűk keresztmetszet tehát jelenleg a webszerver, de csak ennek a komponensnek a fejlesztésével vagy többszörözésével csak korlátozott mértékben növelhető a teljesítmény, mert nagyon hamar a DB2 válik majd szűk keresztmetszetté.

e) Miért nem egyféle foglaltsági időt vettünk figyelembe a két erőforrástípusnál?
Megoldás
Azért, mert mind a DB szerver, mind a WS egy-egy kis rendszer önmagában is, és belül a diszk I/O, ill. a CPU bizonyul szűk keresztemetszetnek jelen esetben. Más rendszerben, más feladatot végrehajtva lehet, hogy az egyik erőforrás hálózati linkje, míg a másik erőforrás RAM sávszélessége fog szerepelni. Vegyük észre, hogy ez egy absztrakció, melynek célja a számítások egyszerűsítése a nem (vagy kevésbé) releváns adatok eltávolításával, ami abból indul ki, hogy az elhanyagolt adatok hatása a megtartott adatokénál jóval kisebb (itt: a webszerver memóriája vagy merevlemez sávszélessége sokkal később telítődne, mint a processzora, de ezt már el sem érjük, ha a processzor miatt vergődik a rendszer). Egyúttal emlékezzünk vissza a 2. feladat b) részére, ahol adatelemzéssel állapítottuk meg a potenciális szűk keresztmetszetet, vagyis a skálázódás és telítődés szempontjából legmeghatározóbb adatot.

f) Hol csal még így is a modell?
Megoldás
Több egyszerűsítéssel is éltünk, pl.
- lineáris skálázódást feltételeztünk, holott a valós rendszerek ennél általában rosszabbul skálázódnak (ráadásul telítődés közelében hajlamosak leromlani),
- nem vettük figyelembe a valódi rendszerben előforduló összes erőforrást (lásd előző feladat),
- feltétleztük, hogy a kéréseket statikus módon elosztva tökéletes terheléselosztást kapunk, holott ez általában nem igaz: az átlagos értékek hosszú távon a számított módon alakulnak, de rövidebb időszakokra nézve egy átlagosnál hosszabb végrehajtási idejű kérés például rövid időre telítésbe viheti a rendszert.

3. Sziget közlekedési hálózata (zh) ∗

Egy sziget lakói minden reggel munkába menet átkelnek a szigetet ölelő tavon. Észak felé híd vezet, dél felé autóskomp. Az irányonként egysávos híd 200 m hosszú, és 60 km/h sebességgel szabad rajta haladni, a követési távolság (hátsó lámpától hátsó lámpáig 30 m) betartása mellett. A négy komphajó egyenként 15 percenként teszi meg a sziget-szárazföld-sziget kört, és így óránként négyen együtt legfeljebb 800 autót tudnak átvinni a szárazföldre.

a) Mekkora a híd átbocsátóképessége (észak felé)?
Megoldás
Little törvényében az átbocsátás szerepel, nem az átbocsátóképesség – de abban a speciális esetben, amikor pont telítve van a rendszer, a kettő megegyezik:
- $N = X \cdot T \rightarrow X = \frac{N}{T}$;
- $N = \frac{200\ \text{m}}{30\ \text{m/kocsi}} = \frac{20}{3}$ kocsi;
- $T = \frac{200\ \text{m}}{60\ \text{km/h}} = \frac{0{,}2\ \text{km}}{60\ \text{km/h}} = \frac{0{,}2}{60}$ h; tehát
- $X = \frac{20/3}{0{,}2/60} = 2000\ \frac{\text{kocsi}}{\text{h}} = X^{max}$.

b) Hány autó fér el egy kompban?
Megoldás
Az előzőhöz hasonlóan Little törvényéből az átbocsátóképesség:
- $N = X \cdot T; X = 800\ \frac{\text{kocsi}}{\text{h}}$;
- $T = 15\ \text{min} = 0{,}25\ \text{h}$;

ekkor $N = 200$, tehát egyszerre 200 autó utazik. Mivel 4 hajó van, ezért egy hajóra 50 kocsi fér fel.

c) A reggeli csúcsforgalomban mekkora a szigetet elhagyó két útvonal együttes átbocsátóképessége?
Megoldás
Az együttes átbocsátóképesség a két átbocsátóképesség összege. A hídon egy irányba óránként

2000 kocsi haladhat át, tehát 2000 $\frac{\text{kocsi}}{\text{h}}$ a híd átbocsátóképessége. A kompok óránként 800 autót visznek át, tehát az átbocsátóképesség 2800 $\frac{\text{kocsi}}{\text{h}}$ egy irányba.

d) Ha délben a szárazföldi főutat baleset miatt lezárták, és a szigeten keresztül (a hídon, majd a kompon átkelve) terelik a forgalmat, mekkora a terelőútvonal átbocsátóképessége?
Megoldás
A terelőút átbocsátóképessége (soros kompozíció): $X = \min(X_{\text{híd}}, X_{\text{komp}}) = 800\ \frac{\text{kocsi}}{\text{h}}$.

e) Valamelyik reggel 7:00 és 8:30 között 900 autó hagyta el a szigetet komppal. Mennyi volt ebben az időszakban a kompok átbocsátása és kihasználtsága?
Megoldás
Átbocsátás: $X = \frac{K}{T} = \frac{900}{1{,}5} = 600\ \frac{\text{kocsi}}{\text{h}}$.
Kihasználtság: $U = \frac{X}{X^{\max}} = \frac{600\ \frac{\text{kocsi}}{\text{h}}}{800\ \frac{\text{kocsi}}{\text{h}}} = 0,75 = 75\%$.

f) A fenti mérésben átlagosan hány autó állt sorba egyszerre a parton, ha az autók jól időzítve, átlagosan fél perccel a beszállásuk előtt érkeztek kompkikötőhöz?
Megoldás
Komphoz sorbanállásra Little-törvény: $N = X \cdot T = 0,5\ \text{min} \cdot 600\ \frac{\text{autó}}{h} = 5$ autó.

4. Tudásbázis (∗)

Vállalatunk nyilvános szakmai tudástára egymásra is hivatkozó szócikkeket kínál a cég termékeit világszerte használó ügyfeleknek. Egyetlen szócikk lekérésének kiszolgálásához a szervert átlagosan 60 ms-ig veszi igénybe. A szócikk megtekintése után az olvasó csak az esetek 30%-ában hagyja el az oldalt, többnyire ugyanis egy újabb szócikkre mutató hivatkozásra kattint.

a) Egy olvasó összes tudásszomjának kielégítéséhez átlagosan mekkora szerveridő szükséges?
Megoldás
Egy szócikk lekérésének kiszolgálása átlagosan 60 ms, egy felhasználó pedig átlagosan $v = \frac{1}{0,3}$ szócikket tekint meg,[2] tehát $T = 60\ \frac{\text{ms}}{\text{szócikk}} \cdot \frac{1}{0,3}\ \frac{\text{szócikk}}{\text{felhasználó}} = 200\ \frac{\text{ms}}{\text{felhasználó}}$. A v most is a vizitációs szám.

b) Tekintsük úgy, hogy az egyes kérések a szerveren nem párhuzamosíthatóak. Óránként hány egyedi látogatót képes kiszolgálni a szerver?
Megoldás
Maximális eset, amikor a kihasználtság 100%, azaz $U = 1$. Ekkor $U = X \cdot T \to X = \frac{U}{T} = \frac{1}{0,2} = 5\ \frac{\text{látogató}}{\text{s}}$. Óránként $3600\ \text{s} \cdot 5\ \frac{\text{látogató}}{\text{s}} = 18000$ látogató.

5. Adatbázis teljesítménymodellezése

Egy adatbázis szervert 15 percig monitorozunk. Ez alatt az idő alatt a szerver processzora 12 percig volt foglalt. Azt figyeltük meg, hogy minden tranzakció általában kétszer használta a processzort, és átlagosan 1 ms ideig használatonként (és ezalatt teljesen lefoglalja a CPU-t, nincs párhuzamosítás). Mekkora a rendszer átbocsátása és áteresztőképessége?
Megoldás
Kis segítség: a rendszer és a komponensek átbocsátóképessége közötti viszonyt írja le:

$$X_k = \frac{C_k}{T} = \frac{C_k}{C_0} \cdot \frac{C_0}{T} = V_k \cdot X_0$$

$$V_k = \frac{C_k}{C_0}$$

A visit ratio (használati arány), azt mondja meg, hogy a rendszer szintű átbocsátás hogy aránylik a komponens átbocsátásához.
Tehát: $T = 15$ perc; foglaltsági idő, busy time: $B_{\text{CPU}} = 12$ perc; Visit Ratio: $V_{\text{CPU}} = 2$ CPU használat/tranzakció; $S_{\text{CPU}} = \frac{1\ \text{ms}}{\textbf{CPU használat}}$
A processzor kihasználtsága: $U_{\text{CPU}} = \frac{B_{\text{CPU}}}{T} = 0,8$
Forced Flow törvény: $X_{\text{CPU}} = V_{\text{CPU}} X_0$

[2] Geometriai eloszlás várható értéke (Wikipédia) `http://hu.wikipedia.org/wiki/Geometriai_eloszlás`

Mi a rendszer átbocsátását keressük, tehát átrendezve $X_0 = \frac{X_{\text{CPU}}}{V_{\text{CPU}}}$, majd pedig a kihasználtság törvényét felhasználva

$$X_0 = \frac{X_{\text{CPU}}}{V_{\text{CPU}}} = \frac{(U_{\text{CPU}}/S_{\text{CPU}})}{V_{\text{CPU}}} = \frac{0{,}8/0{,}001 \text{ s/CPU használat}}{2 \text{ [CPU használat/tranzakció]}} = 400 \text{ tranzakció/s}$$

A rendszer áteresztőképessége: 500 tranzakció/s, hiszen 1 tranzakció 2 ms-ig foglalja a CPU-t.

6. Szálkészlet

Legalább hány aktív szálat kell engedélyeznünk egy webszerveren alkalmazásunknak, ha az egyenletes terhelés melletti teljesítményét nem szeretnénk visszafogni? Szálkorlát nélküli mérésekkel megállapítottuk, hogy egy kérés átlagosan 120 ezredmásodpercet tölt a rendszerben, és a szerver másodpercenként 50 felhasználót szolgál ki.

Megoldás

Little törvényét használva $N = X \cdot R = 50$ felhasználó/s $\cdot$ 0,120 s/felhasználó $= 6$, tehát átlagosan 6 kérés van a rendszerben, tehát 6 szálat kell indítanunk, hogy ne fogjuk vissza a teljesítményt. (Burst esetén nyilván több szálra van szükség.)

7. Közösségi oldal

Internetes közösségi oldalt működtetünk. Az utóbbi időben számottevően népszerűbb lett az oldal, de ezáltal a válaszidő is kellemetlenül megnőtt. Az üzleti cél, hogy csúcsidőszakban egyszerre 1500 felhasználót átlagosan négy másodperces válaszidővel szolgáljon ki a honlap.

a) Minimálisan mekkorára kell tervezni a kiszolgáló infrastruktúra átbocsátóképességét, ha az azon kívüli késleltetés (hálózati forgalom, HTML megjelenítés a kliensoldalon) egy másodpercnek becsülhető?

Megoldás

Tehát a kiszolgáló infrastruktúránknak átlagosan 3 másodperces válaszidővel kell kiszolgálni egyszerre 1500 felhasználót. Little-törvényt alkalmazva: $N = 1500$, $T = 3 \frac{\text{s}}{\text{kérés}}$, tehát $X = \frac{N}{T} = 500 \frac{\text{kérés}}{\text{s}}$

b) Az újratervezett weboldalon a mérések szerint egyetlen kérés kiszolgálása átlagosan 20 ms CPU-időt igényel a webszerveren, és 12,5 ms erejéig foglal le egy adatbázisszervert. Jelenleg 15 webszerver fogadja a kéréseket és az adatbázis 5 kiszolgálóra van replikálva. Lineáris skálázhatóságot feltételezve, milyen számítógépből és mennyit kell még legalább venni, hogy a fenti cél teljesülhessen?

Megoldás

$T_{\text{CPU}} = 20$ ms $= 0{,}02$ s, $T_{\text{DB}} = 12{,}5$ ms $= 0{,}0125$ s. A CPU-nak és adatbázisnak is legalább 500 kérést kell tudnia kiszolgálni másodpercenként, hogy a teljes rendszer is képes legyen erre (akár szekvenciális, akár párhuzamos kompozíciót alkalmazunk). Jelenleg az erőforrások egyetlen példányára: $X_{\text{CPU}}^{\max} = \frac{1}{T_{\text{CPU}}} = 50 \frac{\text{kérés}}{\text{s}}$, $X_{\text{DB}}^{\max} = \frac{1}{T_{\text{DB}}} = 80 \frac{\text{kérés}}{\text{s}}$. Tehát a 15 webszerver átbocsátó képessége együttesen 750 $\frac{\text{kérés}}{\text{s}}$, míg az 5 adatbázis szerveré csak 400 $\frac{\text{kérés}}{\text{s}}$. Tehát még *kell 2 db adatbázis szerver*, hogy az adatbázis réteg elérje a kívánt átbocsátó képességet.

c) (∗) A kibővített rendszerben mekkora lesz az egyes szervertípusok kihasználtsági aránya? Ha az a cél, hogy még a csúcsidőszakban is legfeljebb 50%-os legyen a kihasználtság, meddig kellene még bővíteni a rendszert?

Megoldás

A 15 webszerver átbocsátó képessége együttesen $X_{\text{web}}^{\max} = 750 \frac{\text{kérés}}{\text{s}}$, a csúcsidőszakban a szükséges átbocsátás pedig $X_{\text{web}} = 500 \frac{\text{kérés}}{\text{s}}$. A kihasználtságuk tehát $U_{\text{web}} = \frac{X_{\text{web}}}{X_{\text{web}}^{\max}} = \frac{2}{3}$. Ugyan ezzel a módszerrel: $U_{\text{DB}} = \frac{X_{\text{DB}}}{X_{\text{DB}}^{\max}} = \frac{500}{520} = 0.96$.

Ha 50%-os kihasználtságot szeretnénk, akkor $\frac{X_{\text{web}}}{U_{\text{web}}} \left(= \frac{X_{\text{DB}}}{U_{\text{DB}}}\right) = \frac{500 \frac{\text{kérés}}{\text{s}}}{0.5} = 1000 \frac{\text{kérés}}{\text{s}}$ átbocsátó képességgel kell rendelkeznie az infrastruktúrának csúcsidőszakban. Ehhez 20 webszerver és 13 adatbázis szerver kell.

d) Tekintsünk csak 2 db webszervert és 3 db adatbázis szervert. Készítsünk állapot alapú modell(eke)t, amely(ek) az infrastruktúra erőforrásait modellezi(k) az elérhetőségeik (szabad/fog-

lalt) szerint. Milyen tervezői döntésekkel szembesülünk? Mik az egyes lehetőségek előnyei és hátrányai?

Megoldás

Lehetőségek:

- Az erőforrásokat *típusonként összevonva* modellezzük aszerint, hogy mennyi foglalt belőlük. Tehát lesz egy 0–1–2 állapotláncunk a webszerverekre, valamint egy 0–1–2–3 állapotláncunk az adatbázis szerverekre. Az erőforráskészlet teljes modellje ezek aszinkron szorzata lesz.
  A megoldás *előnye*, hogy egyszerű. Ha például szeretnénk erőforrás foglalást is modellezni, akkor az könnyen megvalósítható kooperáló állapotgépekkel: ha az erőforrás állapotgépe nem az utolsó állapotban van, akkor sikerül a foglalás, és ezzel szinkronban az erőforrás állapotgépe is lép egyet „jobbra” (már eggyel kevesebb erőforrás szabad). Az erőforrás felszabadítás hasonlóan történik.
  A megoldás *hátránya*, hogy nem szolgáltat arról információt, hogy melyik erőforrás példány mikor szabad vagy foglalt, így nem tudunk például pontos kihasználtságot mondani az egyes példányok esetén, csak egy átlagos értéket, ami az összes szervert jellemzi.
- Minden erőforrás *példányt külön modellezünk* egy szabad-foglalt állapotpárral (vagy akár még részletesebben). Tehát annyi állapotgép régiónk lesz, ahány erőforrás példányunk van. Az erőforráskészlet teljes modellje ezek aszinkron szorzata lesz.
  A megoldás *előnye*, hogy konkrét erőforrás példányokra is tudunk például kihasználtságot számolni. Vagy ami még érdekesebb: tudunk erőforrásonként meghibásodást és javítást is modellezni és ennek fényében megnézni az egyes metrikák változását. A meghibásodási és javítási ráták különbözhetnek is az egyes példányok esetén, így lehetőség nyílik heterogén erőforrás kollekció (vagy alkatrész elöregedés) modellezésére is.
  A megoldás hátránya, hogy mostantól a fogyasztók felé is több erőforrás példány látszik, ami például megnehezíti a foglalás modellezését. Egy erőforrás foglalásához meg kell keresni egy szabad erőforrást, majd a végén pontosan azt kell felszabadítani. Ez a szituáció még tovább bonyolódik, ha egy művelethez több erőforrásra is szükség van (holtpont, éheztetés). Ebben az esetben célszerű (és szokás is) bevezetni egy erőforrás menedzser komponenst, amely elrejti ezt a folyamatot a fogyasztóktól.