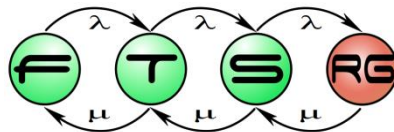


Teljesítménymodellezés

**Budapest University of Technology and Economics
Fault Tolerant Systems Research Group**



Emlékeztető

- Egyensúlyi állapot:
 - Átlagos értékekkel számolunk
 - $\lambda = X$ (érkezési ráta = átboocsátás)
- Átboocsátóképeség:
 - Az elérhető maximális átboocsátás
 - $X_{\max} = \frac{K}{T}$ (K erőforráspéldány esetén)
- Kihasználtság:
 - Az átboocsátás és az átboocsátóképeség aránya
 - $U = \frac{X}{K} \times T$ (K erőforráspéldány esetén)

Alapfogalmak

Terhelési diagram

Erőforrásmodellezés

Folyamatmodellek elemzése

A Little-törvény

FOLYAMATMODELLEK ELEMZÉSE

Hogyan számoljuk ki egy összetett folyamat átbecsátókéességét?

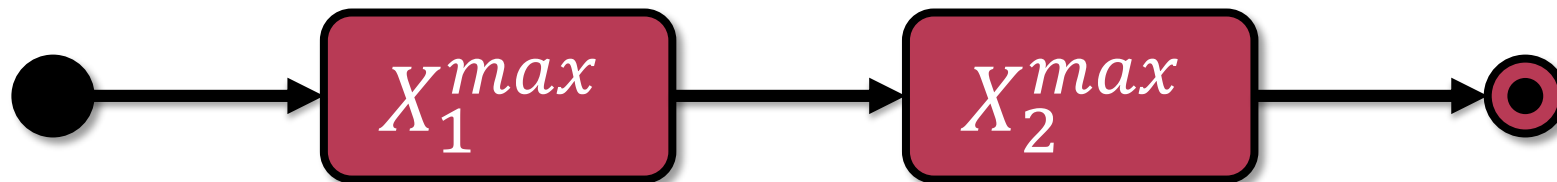
Mit tudunk tervezéskor?

- Általában tevékenységekhez rendelünk erőforrást
 - A tevékenység (átlagos) végrehajtási ideje is adott
 - X_{max} számítható a tevékenységre
- Pl. Neptunban a tárgyfelvétel az adatbáziszszervert terheli 100 ms-ig
 - $T = 100 \text{ ms}$
 - $X_{max} = \frac{1}{T} = 10 \frac{\text{tárgyfelvétel}}{\text{másodperc}}$

Adott folyamatmodell (pl. a felhasználói viselkedés) ismeretében mi a teljes rendszer átbecsátóképessége?

Szekvenciális komponálás

χ^{max}



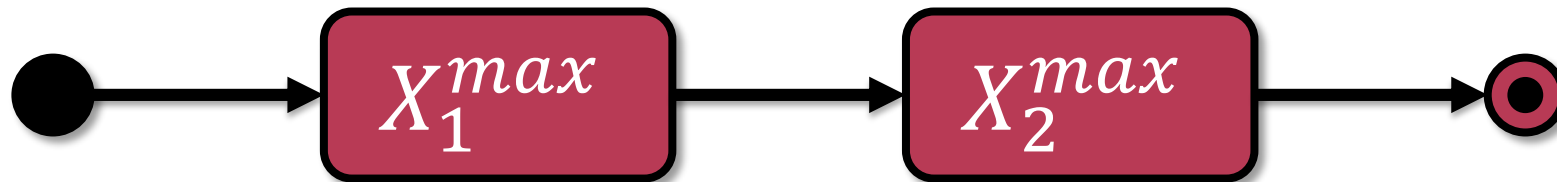
$$\chi^{max} = \min(\chi_1^{max}, \chi_2^{max})$$

Hiába gyors az egyik tevékenység, a tokenek feltorlódnak a másik előtt

Pl. Okmányiroda
Sorszámhúzás (300 db/óra),
Ügyintézés (2 db/óra)

Szekvenciális komponálás

X^{max}

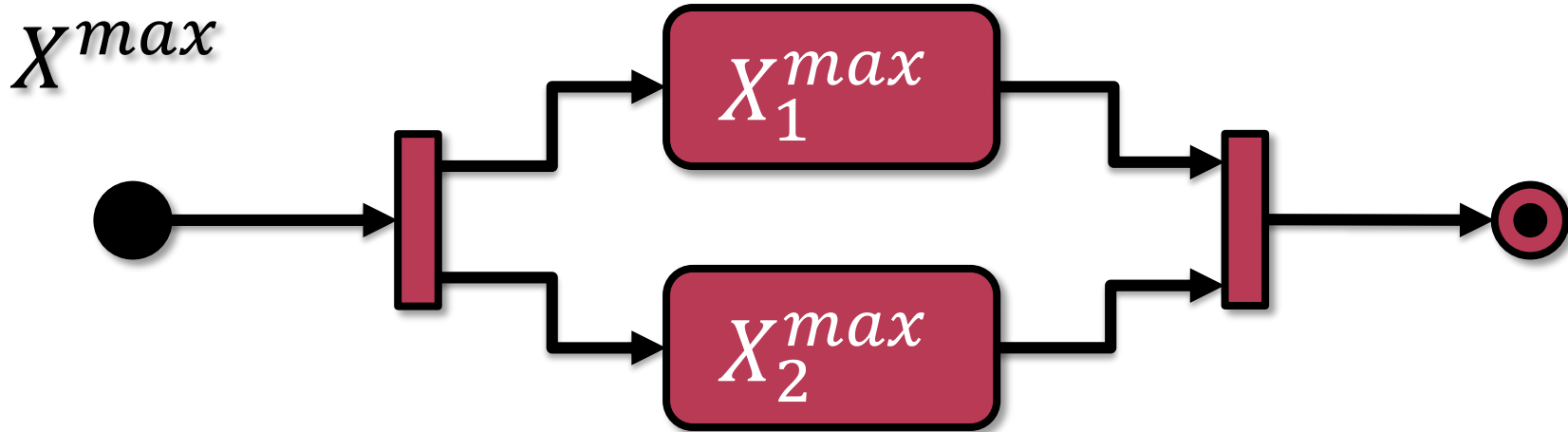


$$X^{max} = \min(X_1^{max}, X_2^{max})$$

Szűk keresztmetszet:

A minimumhelyet adó tevékenység (vagy az ahhoz rendelt erőforrás).

Párhuzamos komponálás

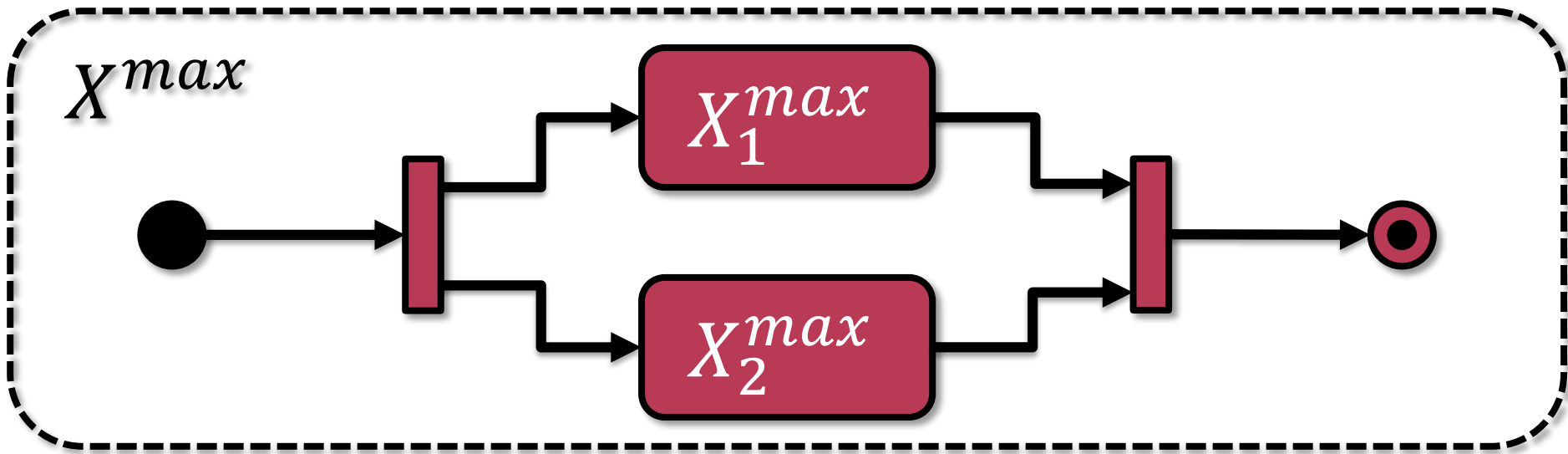


$$X^{max} = \min(X_1^{max}, X_2^{max})$$

Hiába gyors az egyik
tevékenység, a tokeneknek
be kell várniuk egymást

Pl. ZH javítás:
Beugró (30 db/óra),
Nagyfeladat (12 db/óra)

Párhuzamos komponálás



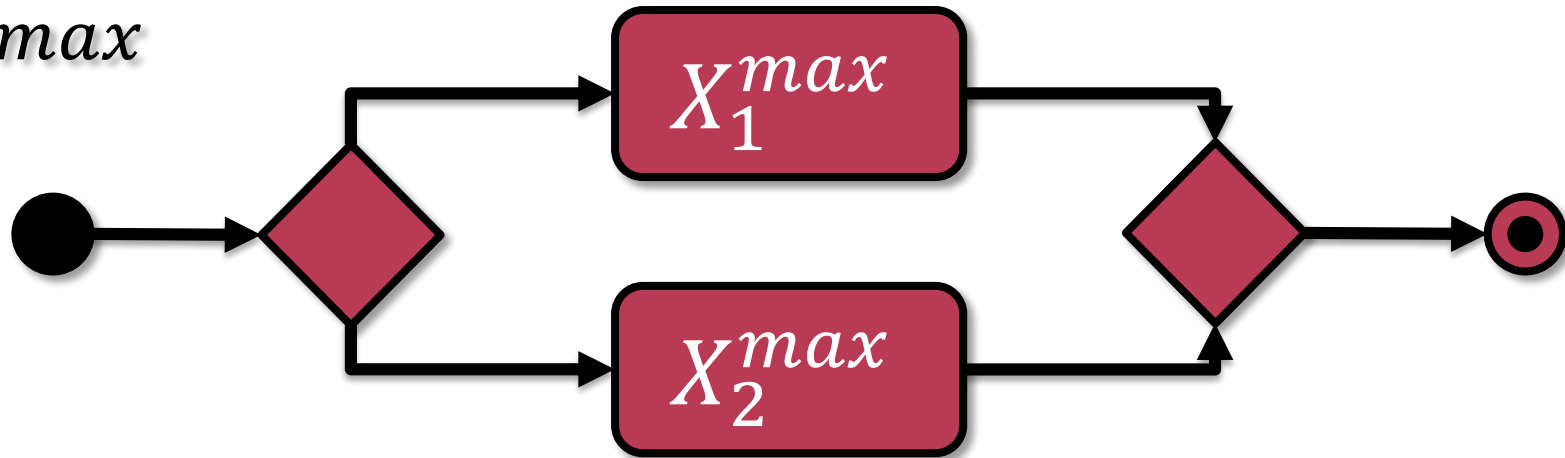
$$X^{max} = \min(X_1^{max}, X_2^{max})$$

Szűk keresztmetszet:

A minimumhelyet adó tevékenység (vagy az ahhoz rendelt erőforrás).

Komponálás szabad választással

χ^{max}



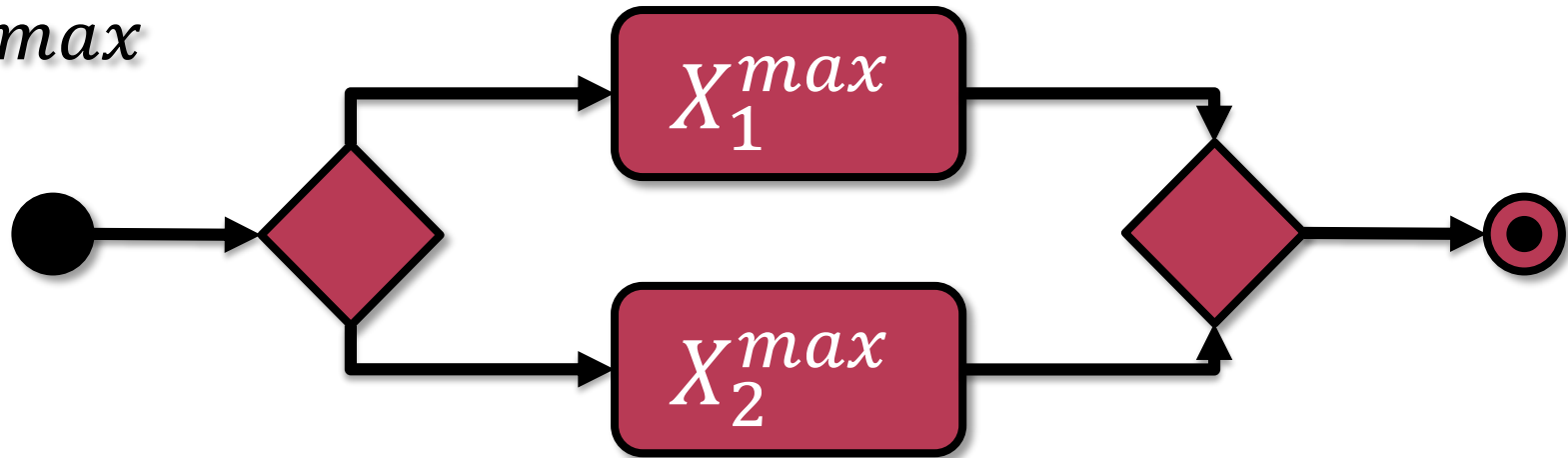
$$\chi^{max} = \chi_1^{max} + \chi_2^{max}$$

A tokenek mindkét irányba mehetnek:
ha az egyik tevékenység telítésben
van, a másik még fogadhat tokenet.

Pl. Áruház:
K db pénztár,
mind 10 db/óra

Komponálás szabad választással

X^{max}

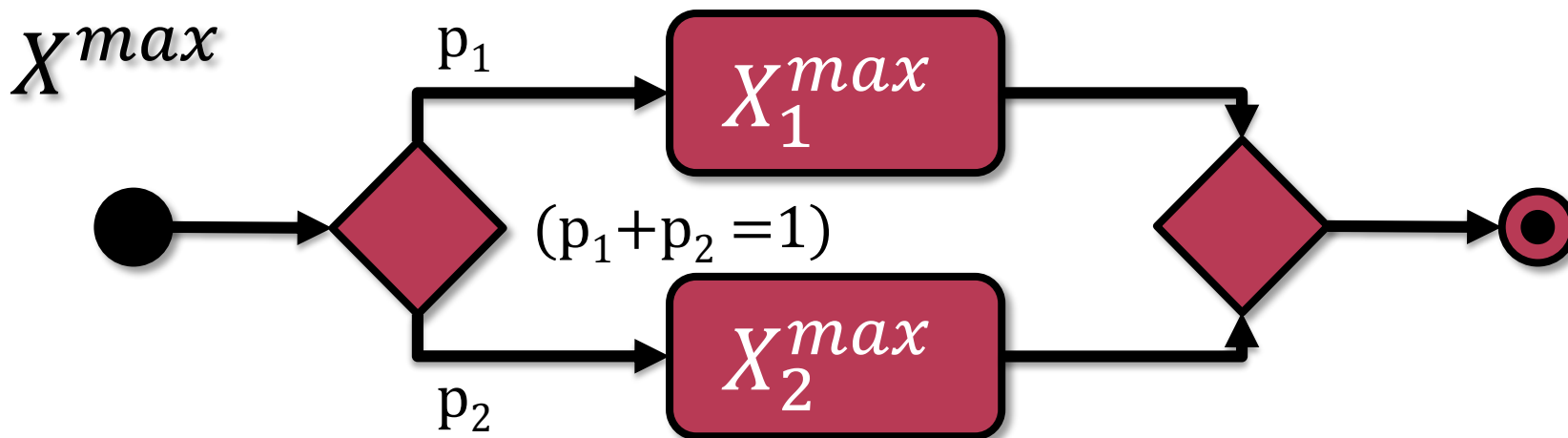


$$X^{max} = X_1^{max} + X_2^{max}$$

Megj: van, amikor a lépéshez rendeljük majd az erőforrások számát, és nem ugyanazt a logikai lépést tüntetjük fel többször a modellben (ld. Szimuláció előadás).

Feltétel: minden erőforráson ugyanannyi ideig tartson

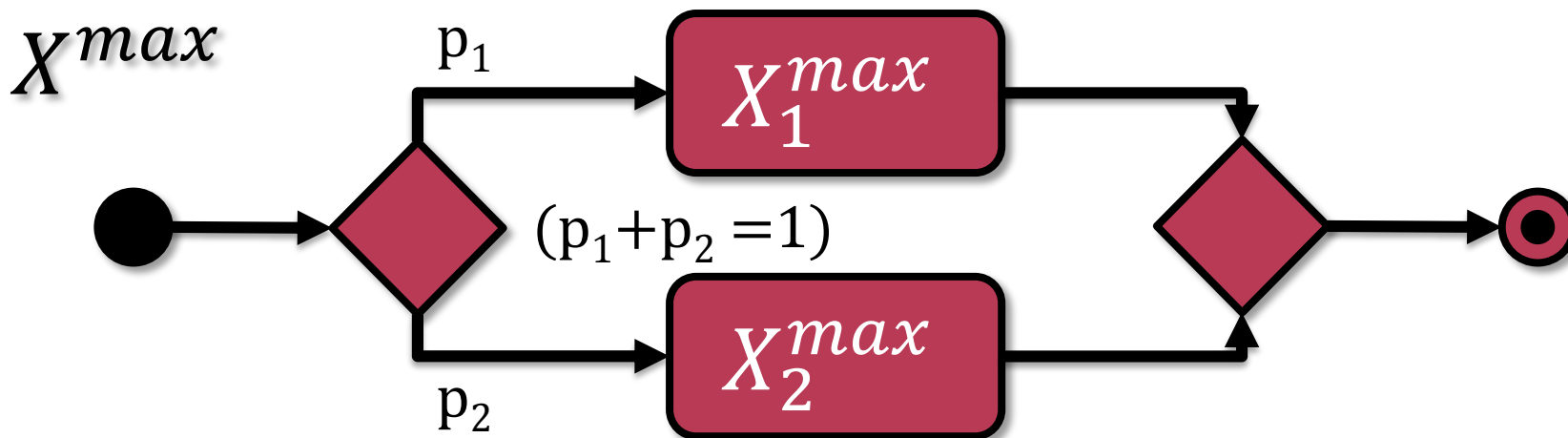
Komponálás kötött arányú választással



$$X^{max} = \min\left(\frac{1}{p_1} \times X_1^{max}, \frac{1}{p_2} \times X_2^{max}\right)$$

A tokennek p_1 és p_2 valószínűséggel választják az első ill. második tevékenységet. A teljes folyamatból tehát $\frac{1}{p_1}$ ill. $\frac{1}{p_2}$ tokenből egy jut az első ill. második tevékenységre.

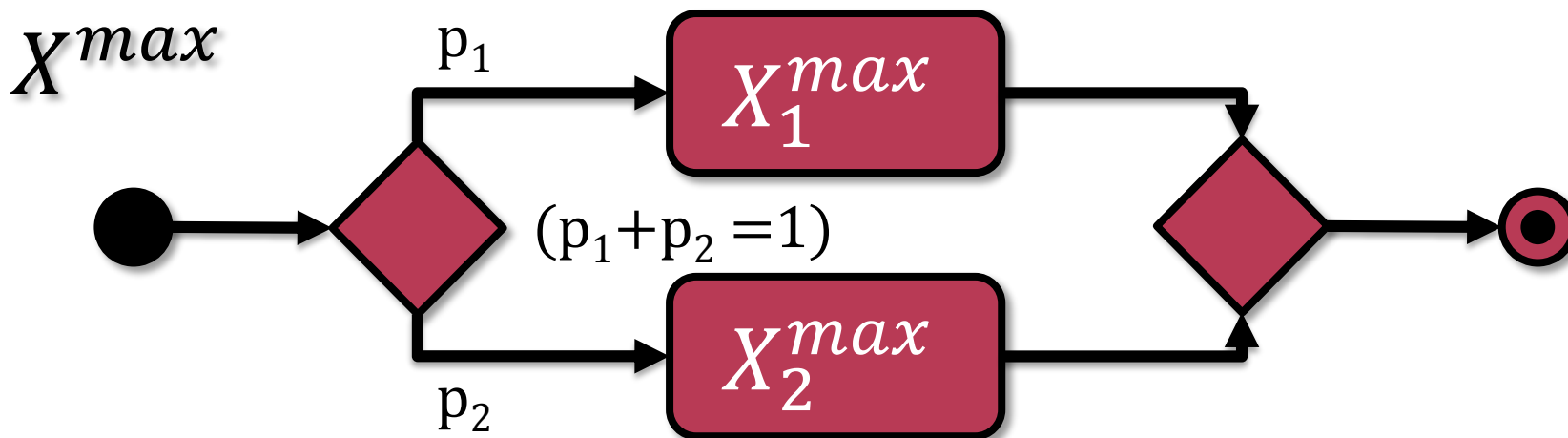
Komponálás kötött arányú választással



$$X^{max} = \min\left(\frac{1}{p_1} \times X_1^{max}, \frac{1}{p_2} \times X_2^{max}\right)$$

Pl. Felhasználó viselkedése egy weblapon:
20% eséllyel vásárol (20 db/s),
80% eséllyel elvet (200 db/s)

Komponálás kötött arányú választással



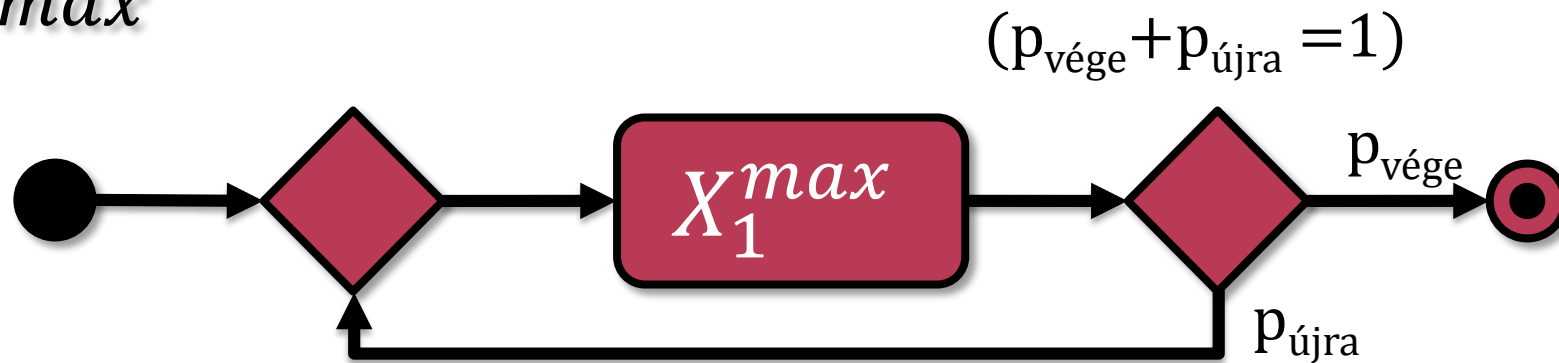
$$X^{max} = \min\left(\frac{1}{p_1} \times X_1^{max}, \frac{1}{p_2} \times X_2^{max}\right)$$

Szűk keresztmetszet:

A minimumhelyet adó tevékenység (vagy az ahhoz rendelt erőforrás).

Komponálás ciklussal

χ^{max}

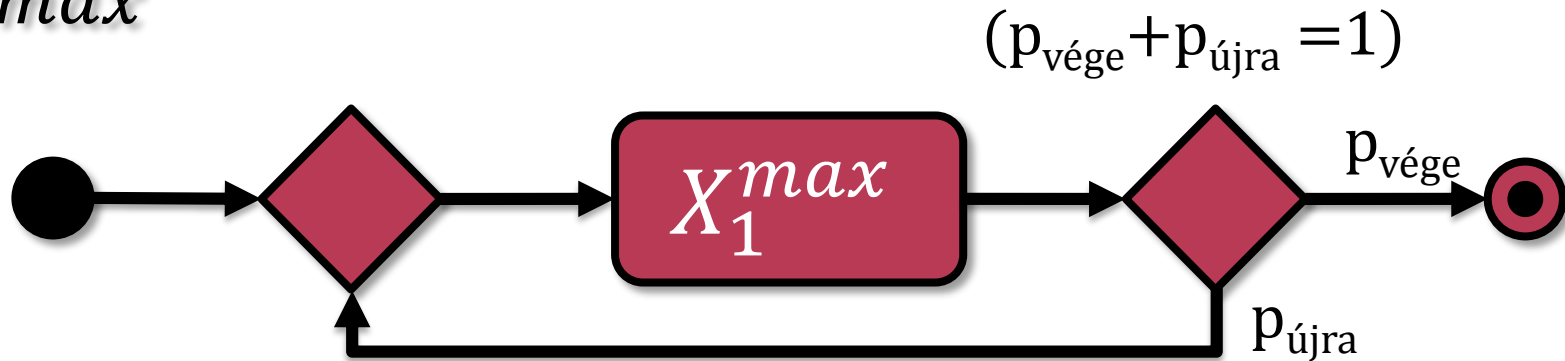


$$\chi^{max} = \frac{1}{p_{vége}} \times \chi_1^{max} = p_{újra} \times \chi_1^{max}$$

Az $\frac{1}{p_{vége}}$ érték az iterációk várható száma
(lásd később a Valószínűségszámítás tantárgyban).

Komponálás ciklussal

χ^{max}



$$\chi^{max} = \frac{1}{p_{vége}} \times \chi_1^{max} = p_{vége} \times \chi_1^{max}$$

Pl. Felhasználó a rendszerben:
10% eséllyel kilép, 90% eséllyel új kérés
15 kérés/s, átlagosan 10 kérés/munkamenet

Vizitációs szám

Választás:
$$X^{max} = \min\left(\frac{1}{p_1} \times X_1^{max}, \frac{1}{p_2} \times X_2^{max}\right)$$

Ciklus:
$$X^{max} = \frac{1}{p_{vége}} \times X_1^{max} = p_{vége} \times X_1^{max}$$

- **Vizitációs szám:** megmutatja, hogy a folyamat végrehajtása során átlagosan hányszor fut le az adott tevékenység/alfolyamat.
 - Választás esetén maga a döntési valószínűség
 - Ciklus esetén a várható iterációk száma

Vizitációs szám

Átbocsátóképesség a vizitációs szám ismeretében:

$$X^{max} = \frac{1}{v} \times X_1^{max}$$

- **Vizitációs szám:** megmutatja, hogy a folyamat végrehajtása során átlagosan hányszor fut le az adott tevékenység/alfolyamat.
 - Választás esetén maga a döntési valószínűség
 - Ciklus esetén a várható iterációk száma

Vizitációs szám

Átbocsátóképesség a vizitációs szám ismeretében:

$$\frac{1}{\chi^{max}} = \nu \times \frac{1}{\chi_1^{max}}$$

- **Vizitációs szám:** megmutatja, hogy a folyamat végrehajtása során átlagosan hányszor fut le az adott tevékenység/alfolyamat.
 - Választás esetén maga a döntési valószínűség
 - Ciklus esetén a várható iterációk száma

Vizitációs szám

Végrehajtási idő a vizitációs szám ismeretében:

$$T_{folyamat} = v \times T_{taszk}$$

- **Vizitációs szám:** megmutatja, hogy a folyamat végrehajtása során átlagosan hányszor fut le az adott tevékenység/alfolyamat.
 - Választás esetén maga a döntési valószínűség
 - Ciklus esetén a várható iterációk száma

Alapfogalmak

Terhelési diagram

Erőforrásmodellezés

Folyamatmodellek elemzése

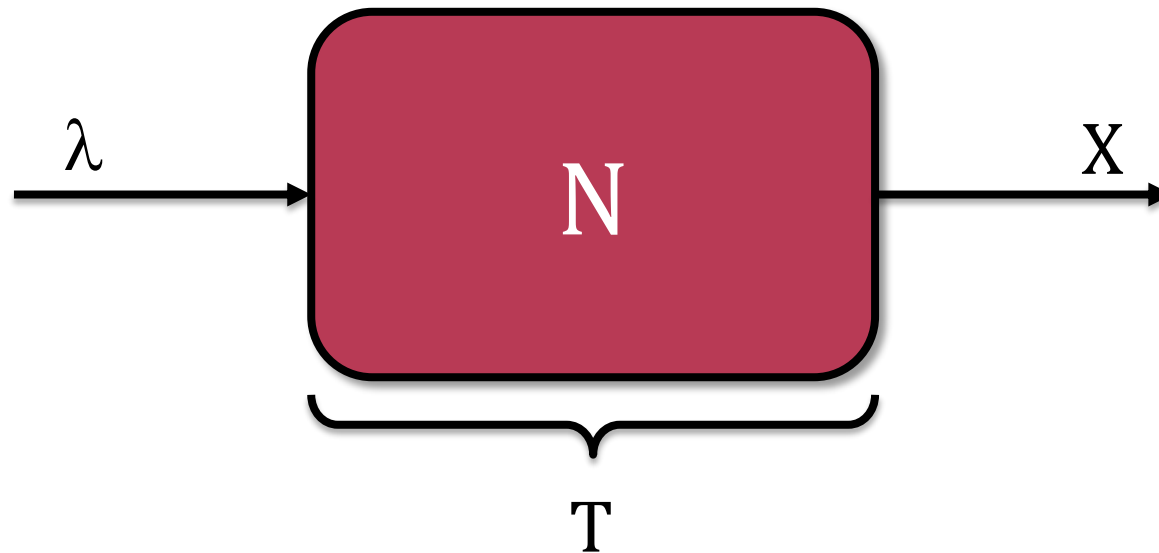
A Little-törvény

A LITTLE-TÖRVÉNY

Avagy az alapképlet

A Little-törvény

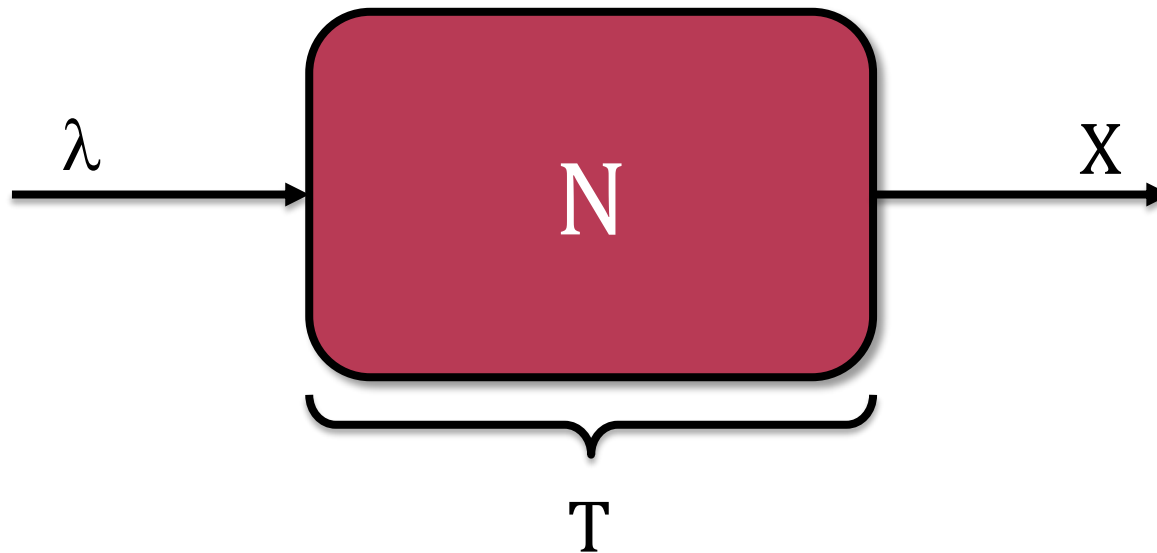
- λ : érkezési ráta
- X : átbecsátás
- T : rendszerben töltött idő
- N : rendszerben lévő tokenek száma



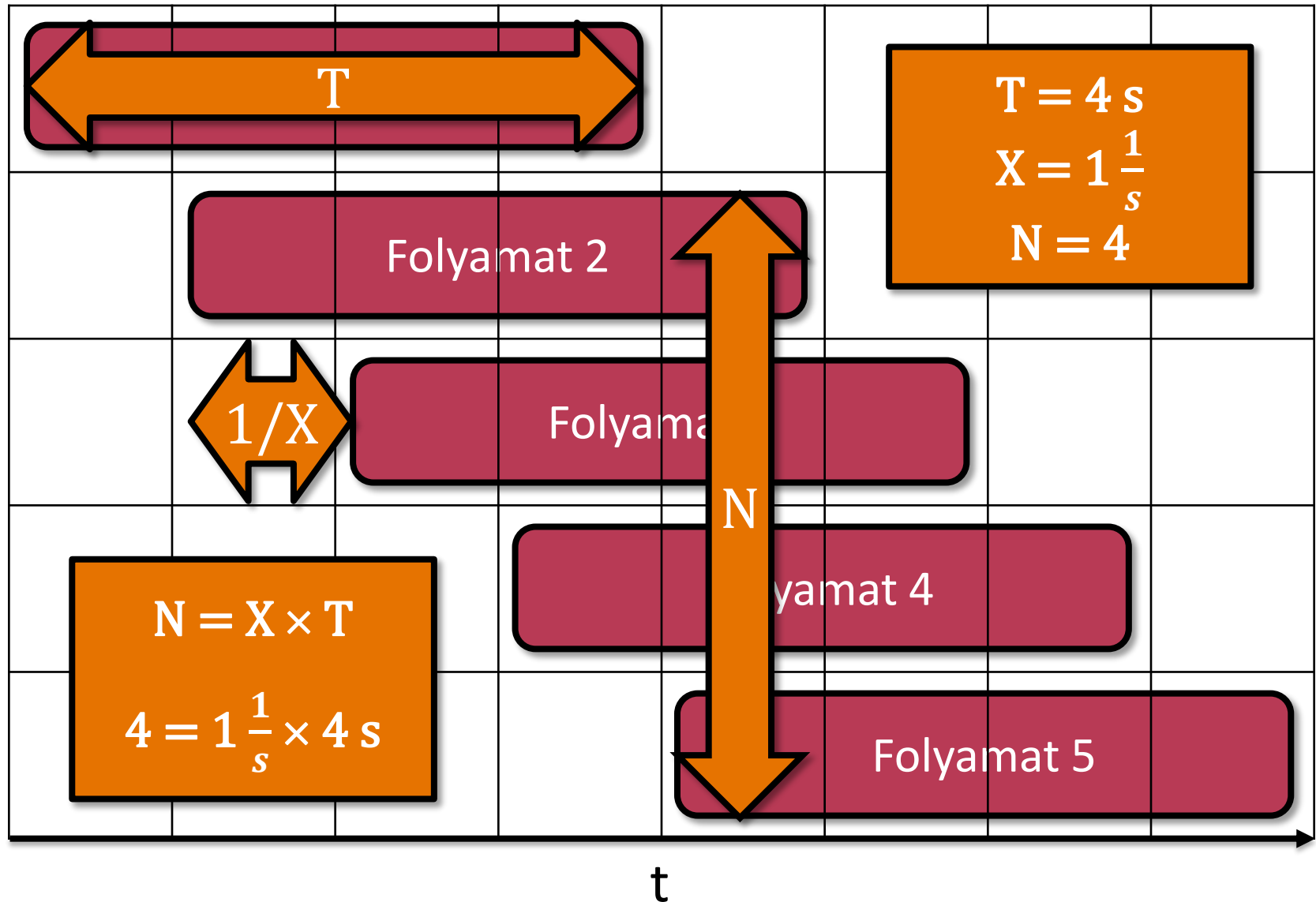
A Little-törvény

- Egyensúlyi állapotban ($\lambda = X$) igaz a Little-törvény:

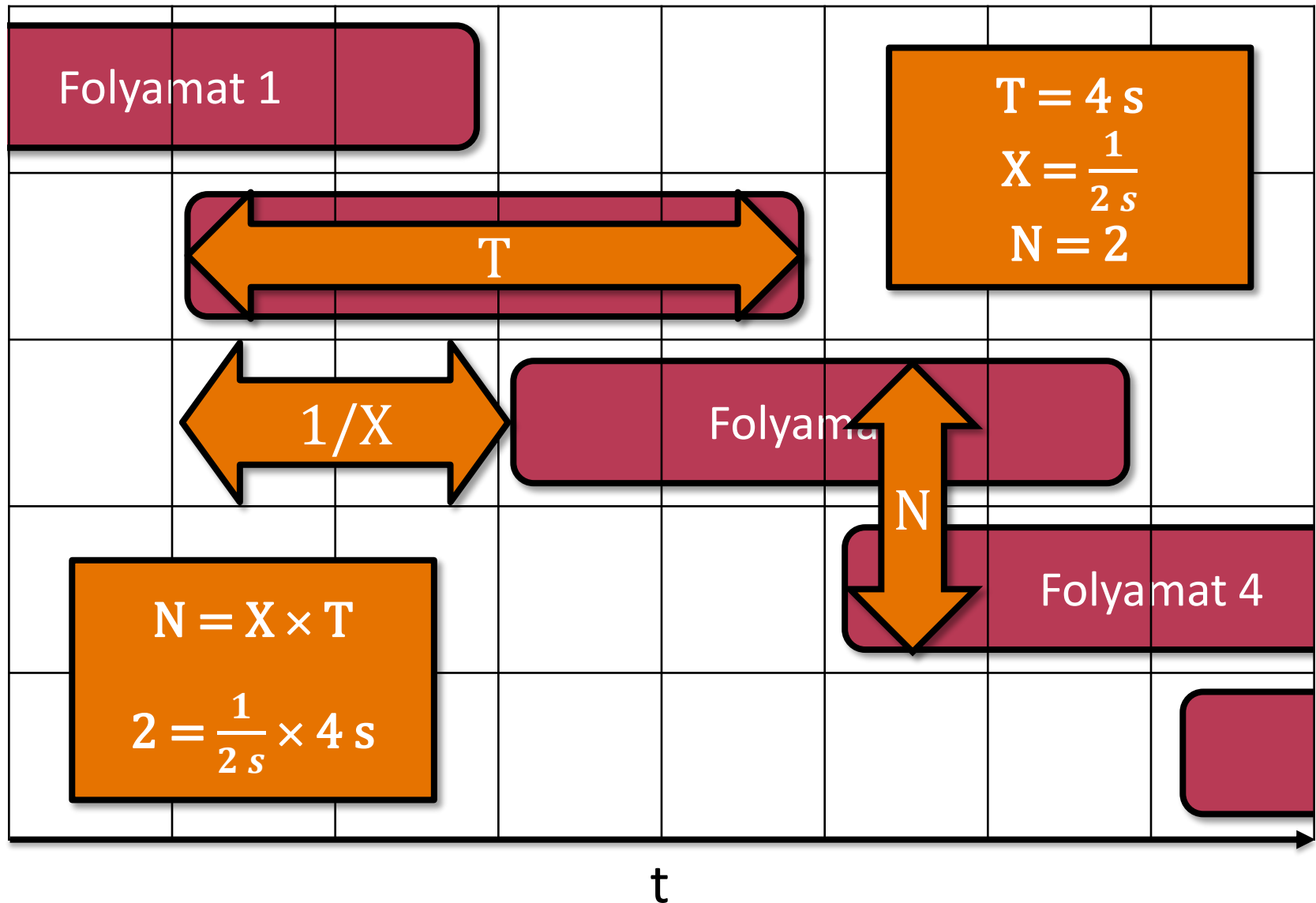
$$N = X \times T$$



A Little-törvény szemléltetése



A Little-törvény szemléltetése



Kihasználtság és a Little-törvény

- K erőforráspéldány: maximum K folyamatpéldány állhat végrehajtás alatt
- A Little-törvény megadja a végrehajtás alatt álló folyamatpéldányok számát (N)
- Az eddigiekből levezethető:

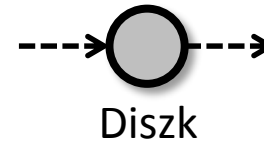
$$U = \frac{X}{K} \times T = \frac{X \times T}{K} = \frac{N}{K}$$

Kihasználtság K
erőforráspéldányra

Little-törvény
($N = X \times T$)

LITTLE TÖRVÉNY: GYAKORLATI PÉLDÁK

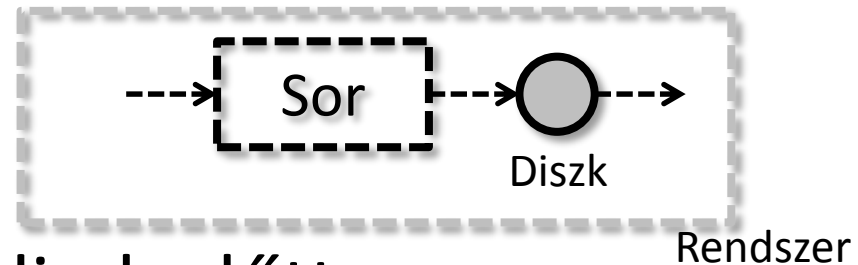
Példa



- Erőforrás: diszk
- 40 kérést szolgál ki másodpercenként (nincs átlapolódás)
- 1 kérés kiszolgálása átlagosan 0,0225 másodpercig tart
- Mekkora a kihasználtság?

$$U = X \times T_{\text{diszk}} = 40 \text{ kérés/mp} \times 0,0225 \text{ mp} = 90\%$$

Példa

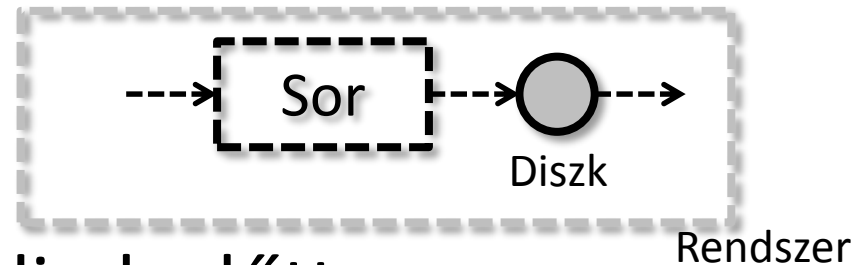


- Sorban állás is van a diszk előtt
- A diszk 40 kérést szolgál ki másodpercenként
- Kérések átlagos száma a rendszerben: 4

Átlagos rendszerben tartózkodási idő? (Trendszer)

Átlagos sorban állási idő? (Tvárakozás)

Példa



- Sorban állás is van a diszk előtt
- A diszk 40 kérést szolgál ki másodper
- Kérések átlagos száma a rendszerbe

Sorbanállási és
diszk kiszolgálási
idő

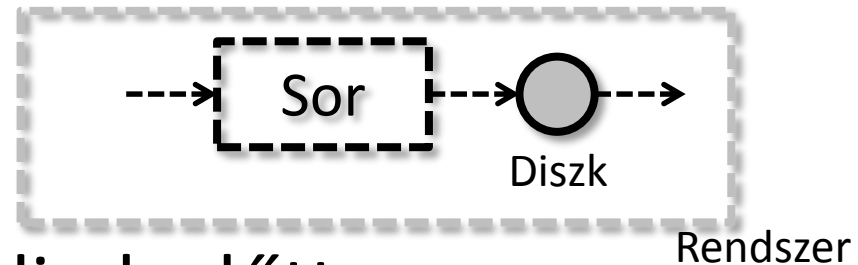
Rendszer

$$N = X \times T \rightarrow T_{rendszer} = 4 \text{ kérés} / 40 \text{ kérés/mp} = 0,1 \text{ mp}$$

Átlagos sorban állási idő?

$$(T_{rendszer} - T_{diszk}) 0,1 \text{ mp} - 0,0225 \text{ mp} = 0,0775 \text{ mp}$$

Példa



- Sorban állás is van a diszk előtt
- A diszk 40 kérést szolgál ki másodpercenként
- Kérések átlagos száma a rendszerben: 4

Kérések átlagos száma a sorban?
($N_{\text{rendszer}} - N_{\text{diszk}}$)
4 kérés – 0,9 kérés = 3,1 kérés

Little törvény a gyakorlatban

■ Szimuláció

- Dobson&Shumsky
- http://ited.informs.org/Vol7No1/DobsonShumsky/security_simulation.php
- <https://youtu.be/UjzXQPGBaNA>

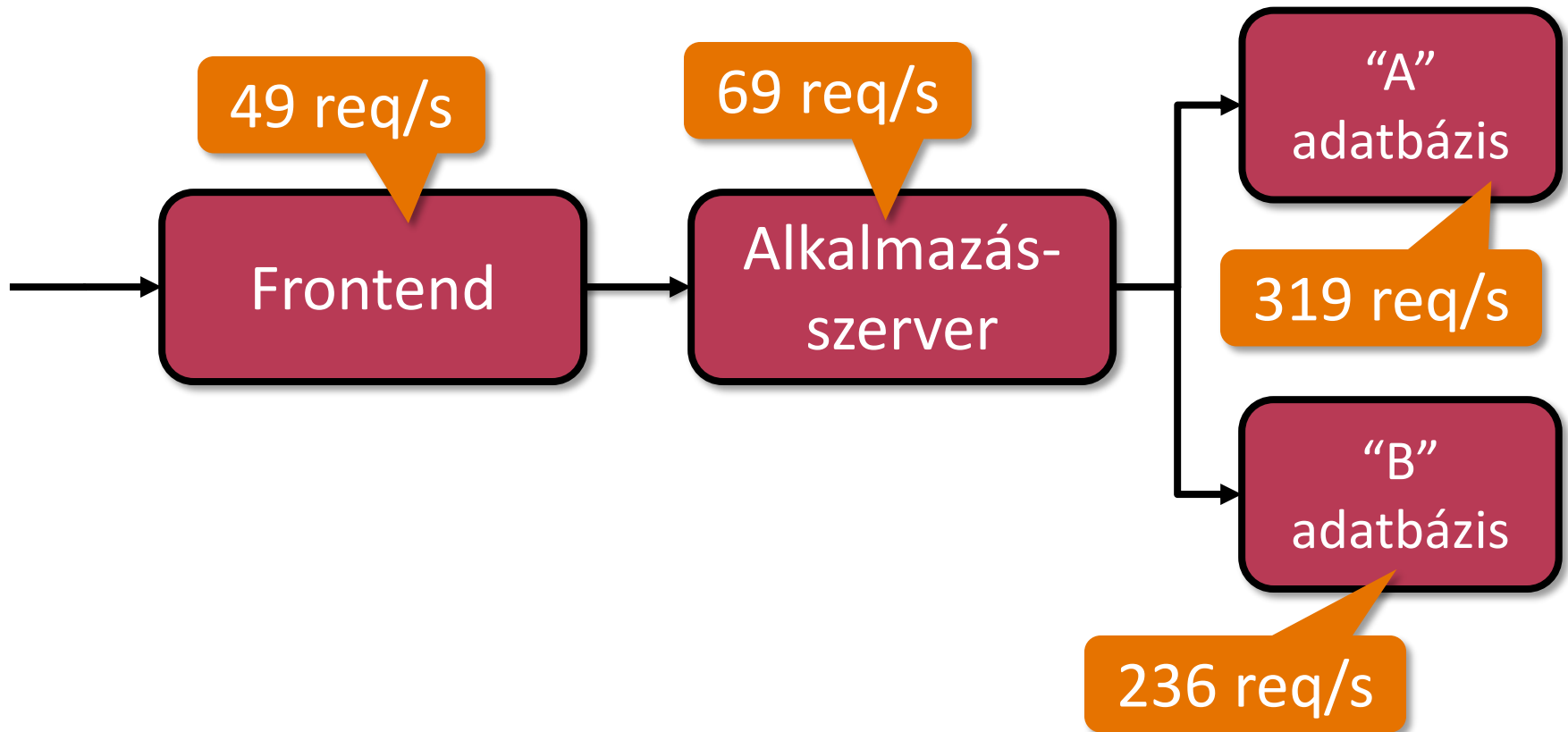
■ Miért oktatják(juk)

- <http://pubsonline.informs.org/doi/pdf/10.1287/ited.7.1.106>

■ Példák

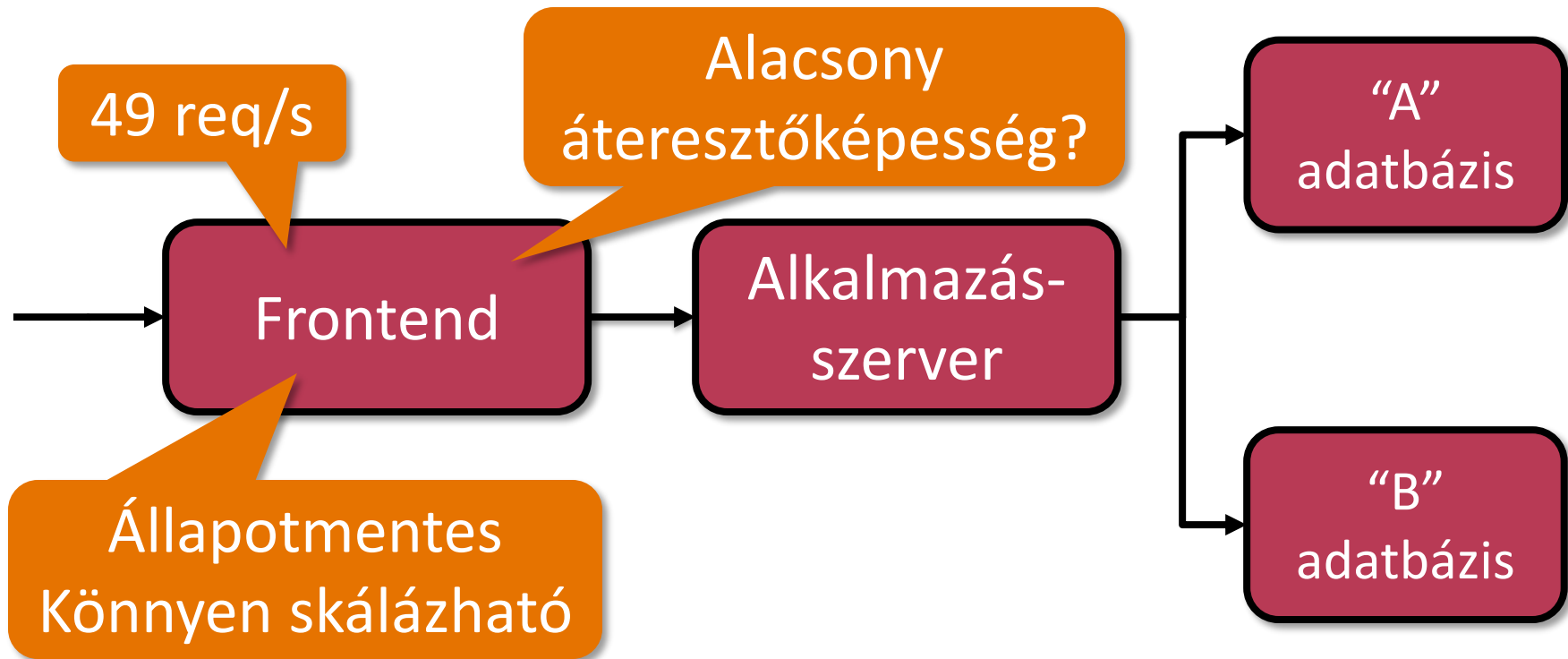
- <http://web.mit.edu/sgraves/www/papers/Little's%20Law-Published.pdf>

Teljesítmény 3 rétegű architektúrában



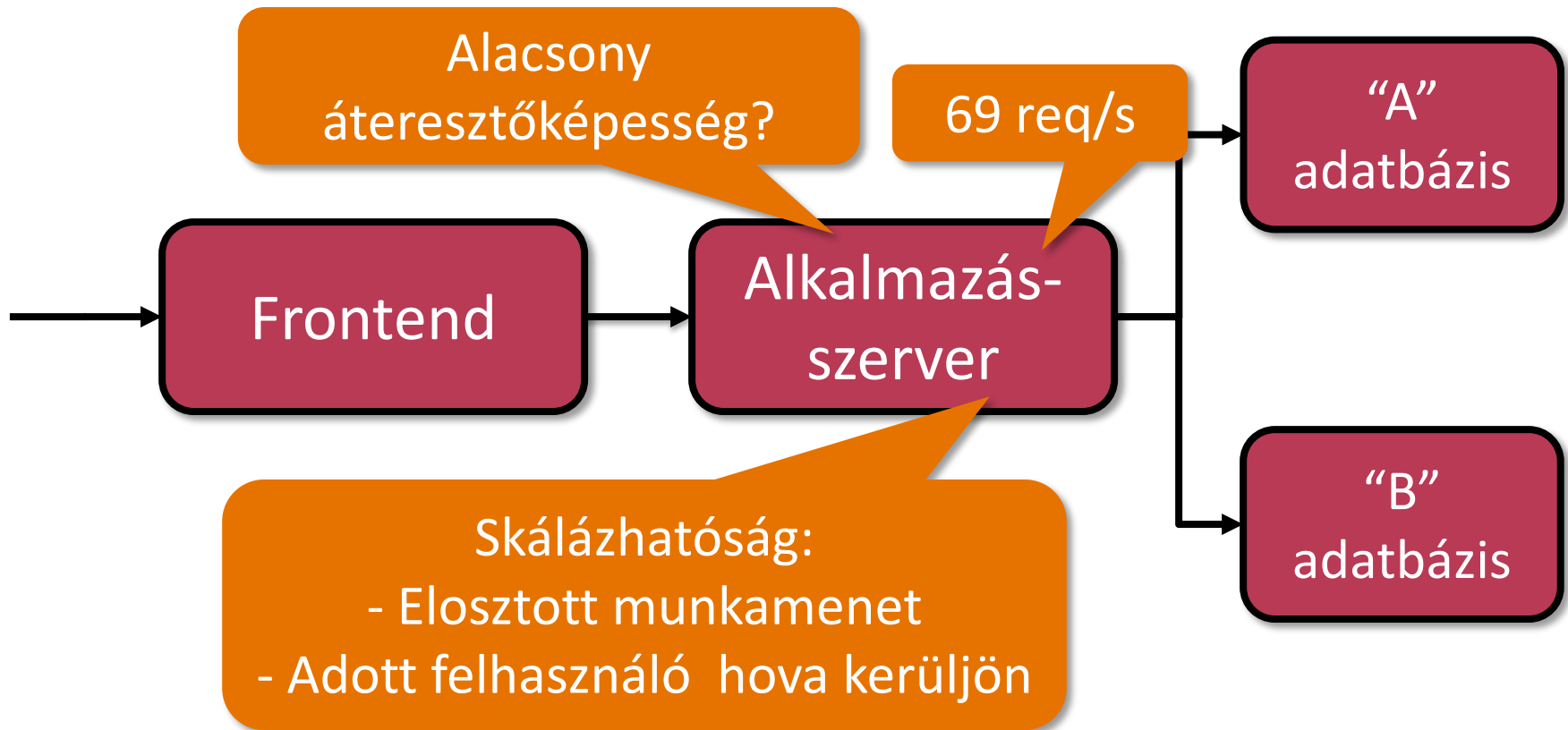
A mérőszámok itt az egész rendszerre érkező terhelésre utalnak! Például az „A. adatbázis” akkor válik szűk keresztmetszetté, ha a rendszerbe 319 lekérdezés érkezik másodpercenként.

Teljesítmény 3 rétegű architektúrában



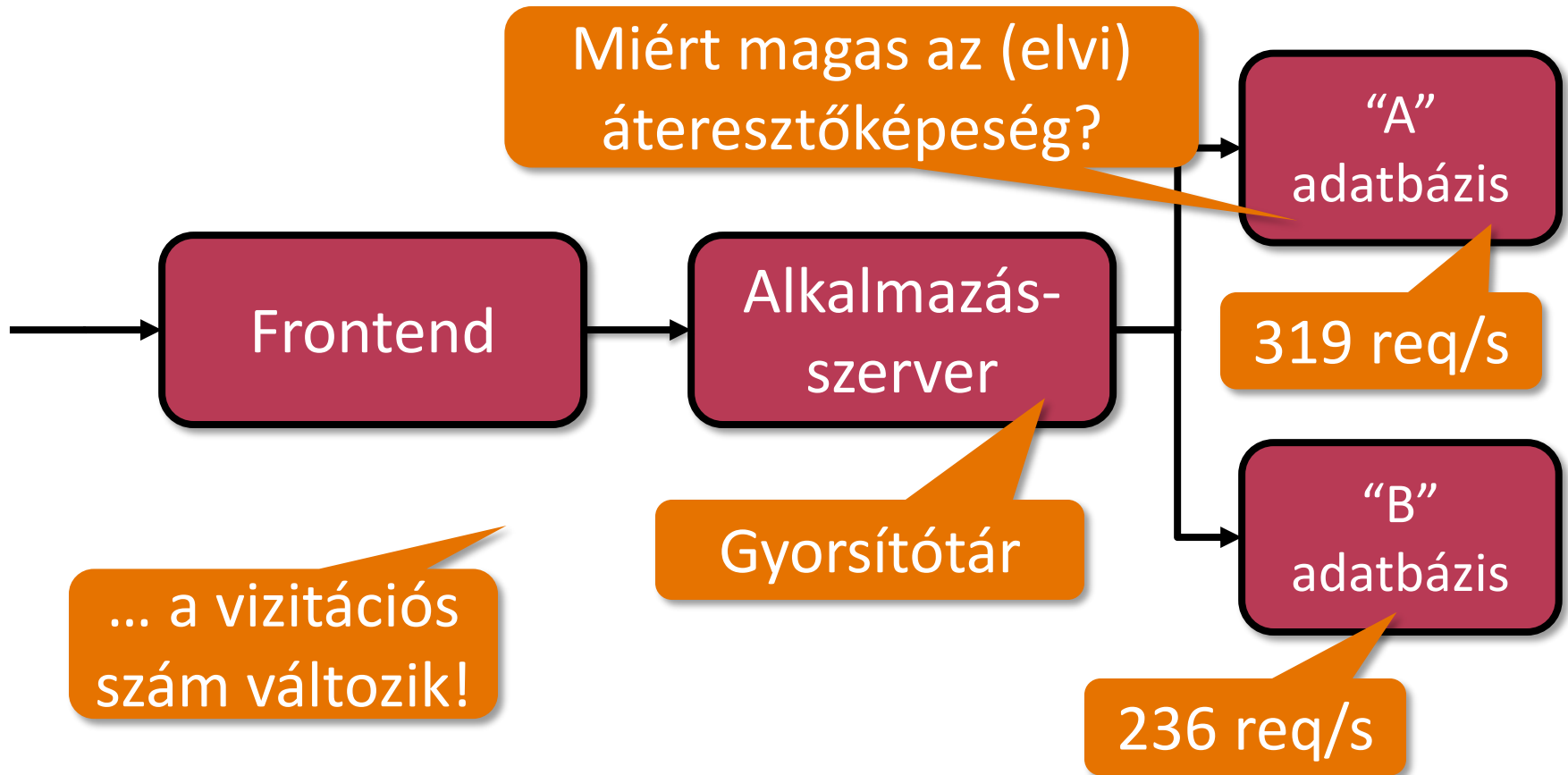
A mérőszámok itt az egész rendszerre érkező terhelésre utalnak! Például az „A. adatbázis” akkor válik szűk keresztmetszetté, ha a rendszerbe 319 lekérdezés érkezik másodpercenként.

Teljesítmény 3 rétegű architektúrában



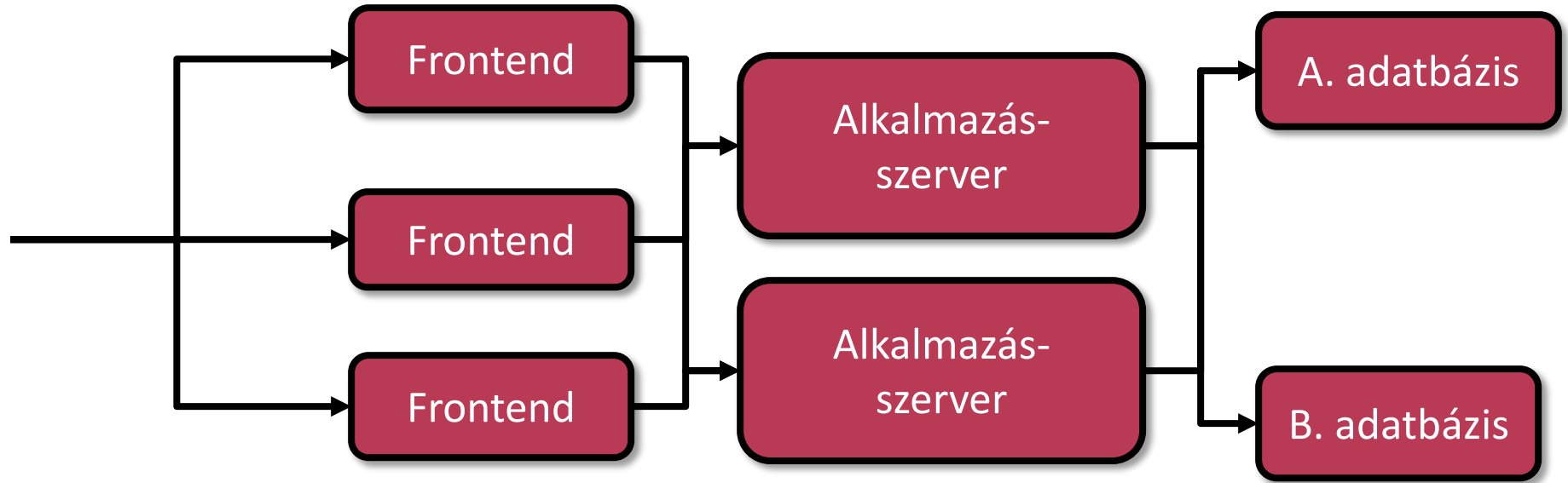
A mérőszámok itt az egész rendszerre érkező terhelésre utalnak! Például az „A. adatbázis” akkor válik szűk keresztmetszetté, ha a rendszerbe 319 lekérdezés érkezik másodpercenként.

Teljesítmény 3 rétegű architektúrában



A mérőszámok itt az egész rendszerre érkező terhelésre utalnak! Például az „A. adatbázis” akkor válik szűk keresztmetszetté, ha a rendszerbe 319 lekérdezés érkezik másodpercenként.

3 rétegű architektúra a valóságban



(Példa: Technológiai háttér érdeklődőknek:)

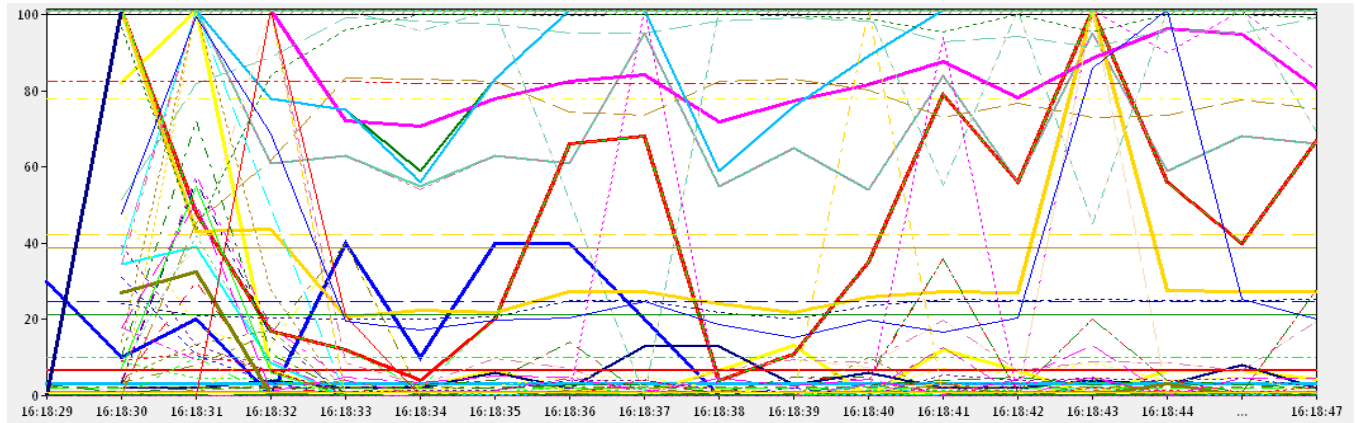
<http://www.projectclearwater.org/wp-content/uploads/2013/05/Clearwater-Deployment-Sizing-10-Apr-13.xlsx>

<http://www.projectclearwater.org/technical/clearwater-performance/>

Mit mérjük/mi a lényeges?

■ Metrikák “kicsiben”

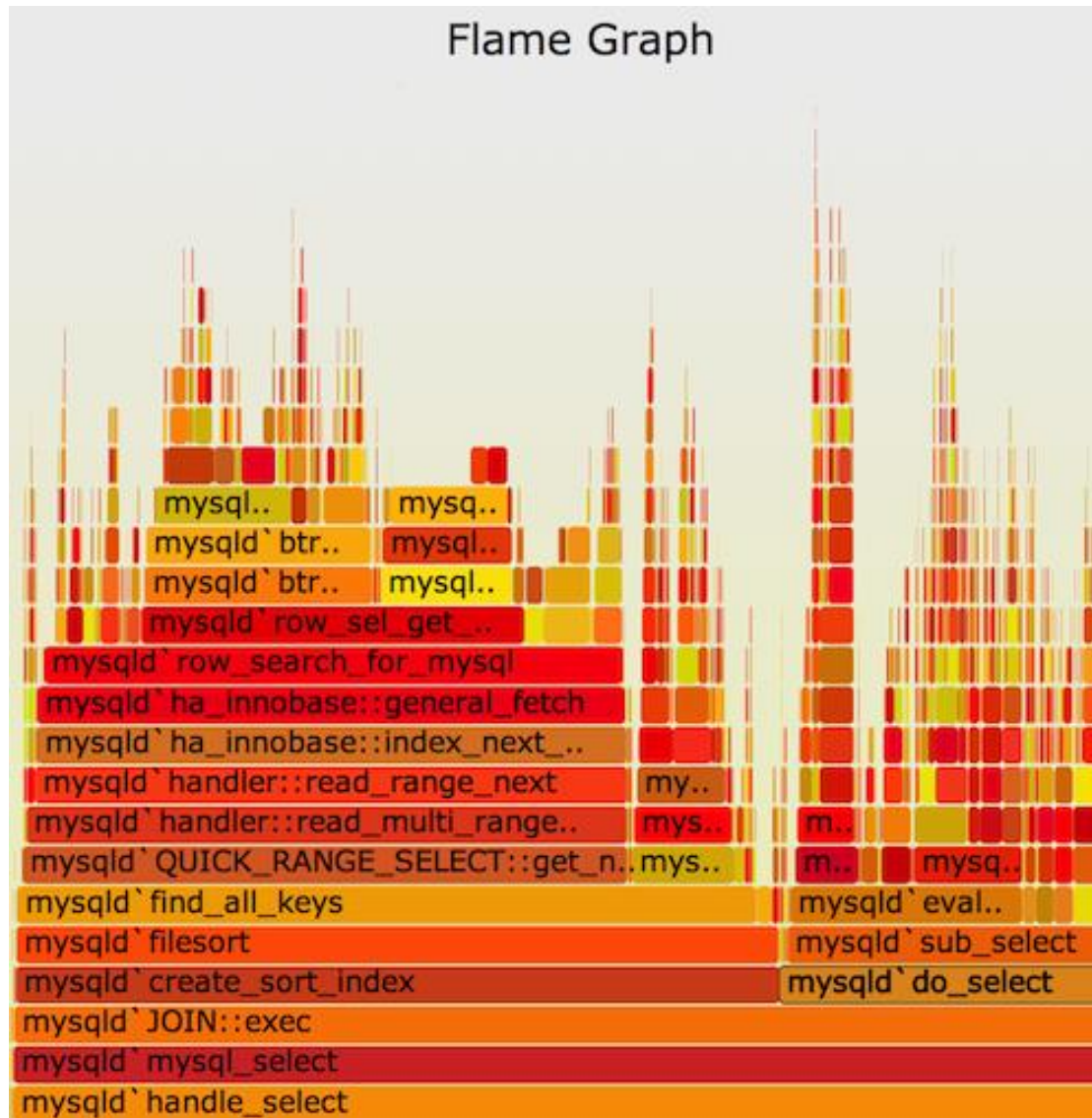
- Pl. Task manager, Resource monitor, ugyanez szerver oldalon....



■ Metrikák “nagyban”

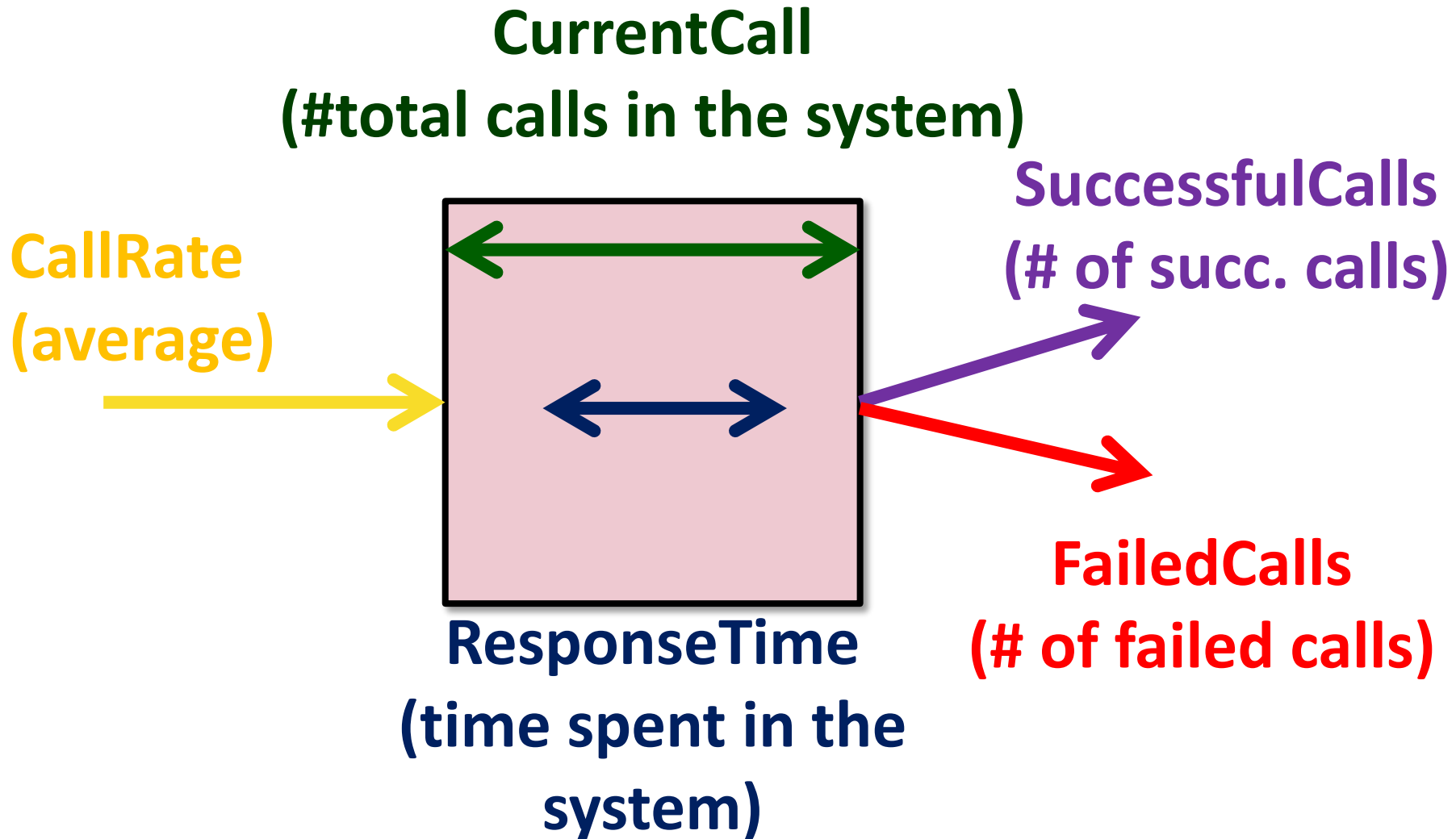
- Pl. virtualizált rendszer
- Melyik az érdekes?

Példa: mit számol ez a gép enyit?

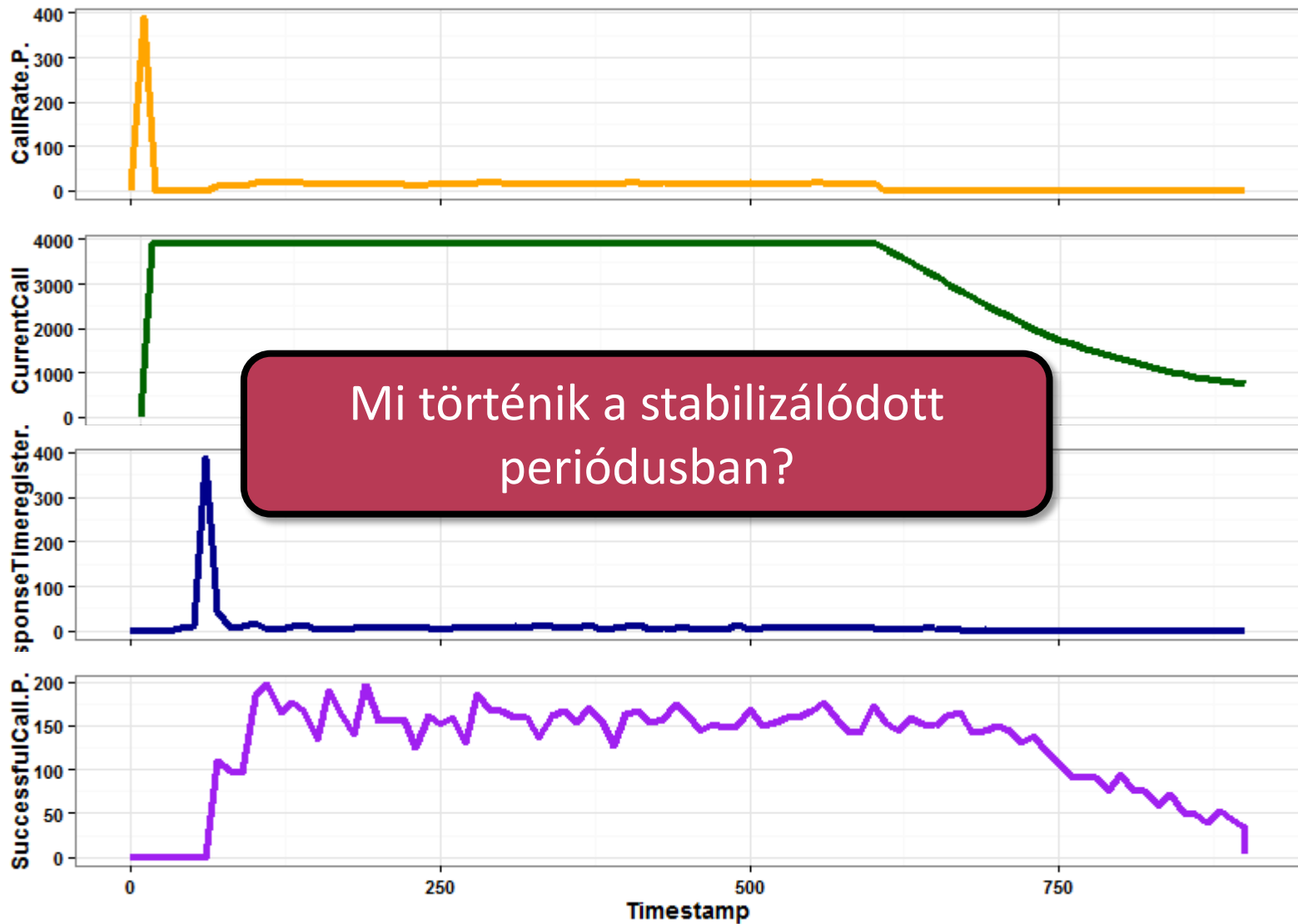


<http://www.brendangregg.com/flamegraphs.html>

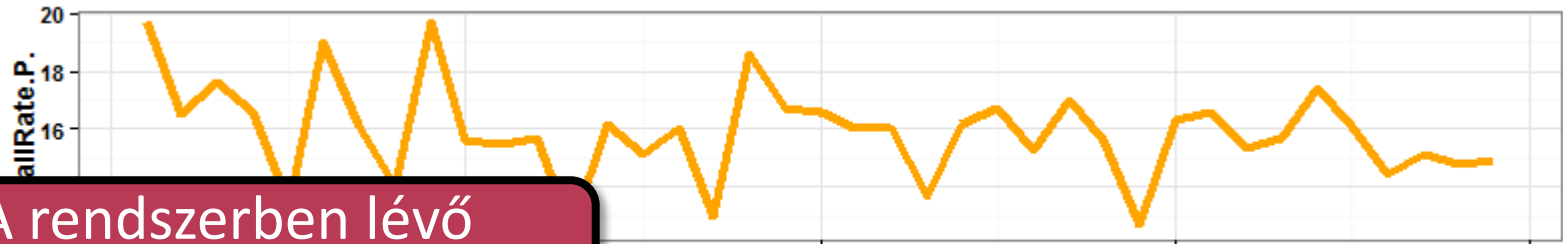
Little törvény példa



Visszacsatolós terhelés



A mérés állandósult állapotban

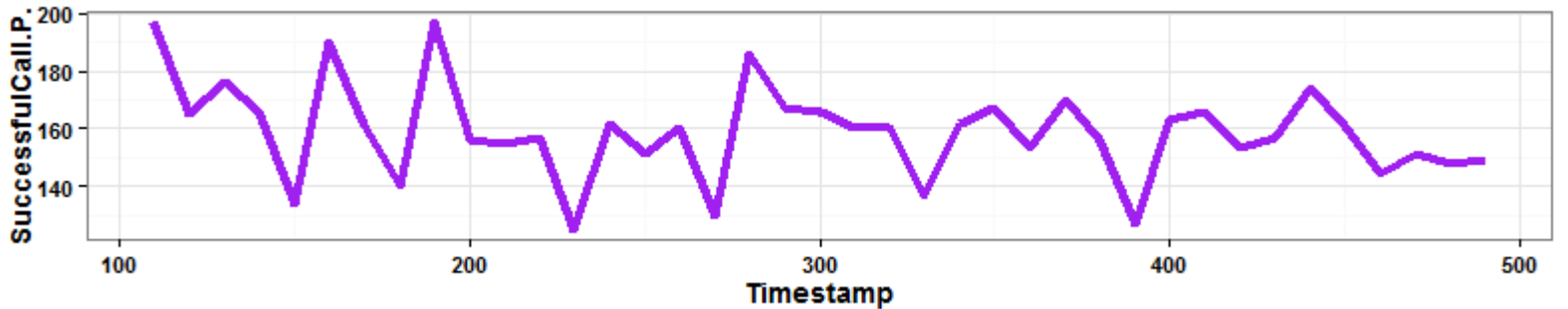
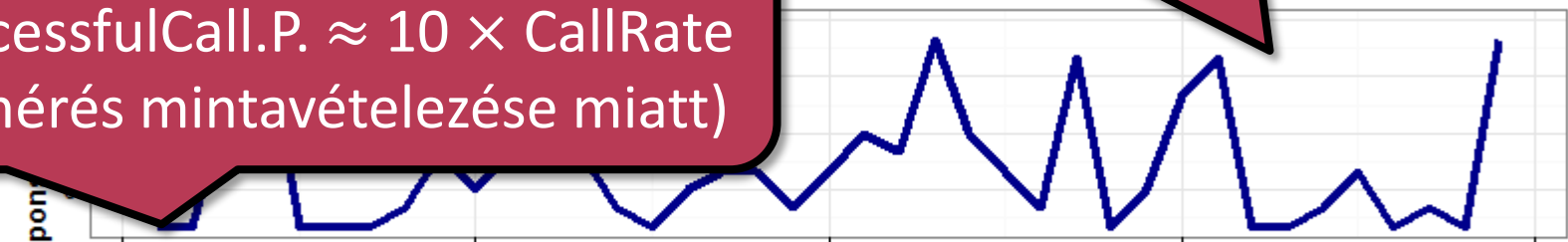


A rendszerben lévő
hívások száma konstans

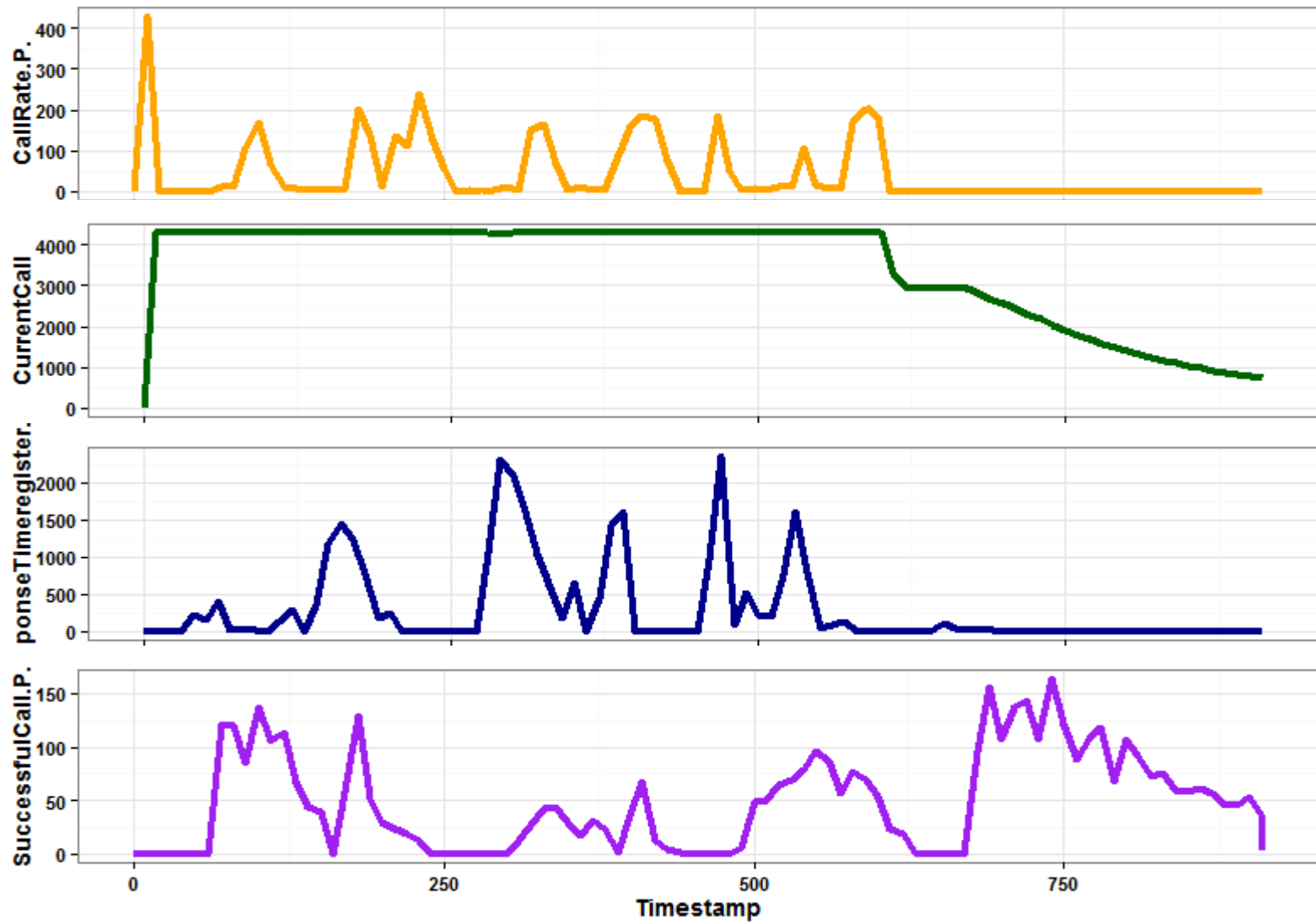
CallRate.P.

A kiszolgálási idő kb. 10 ms

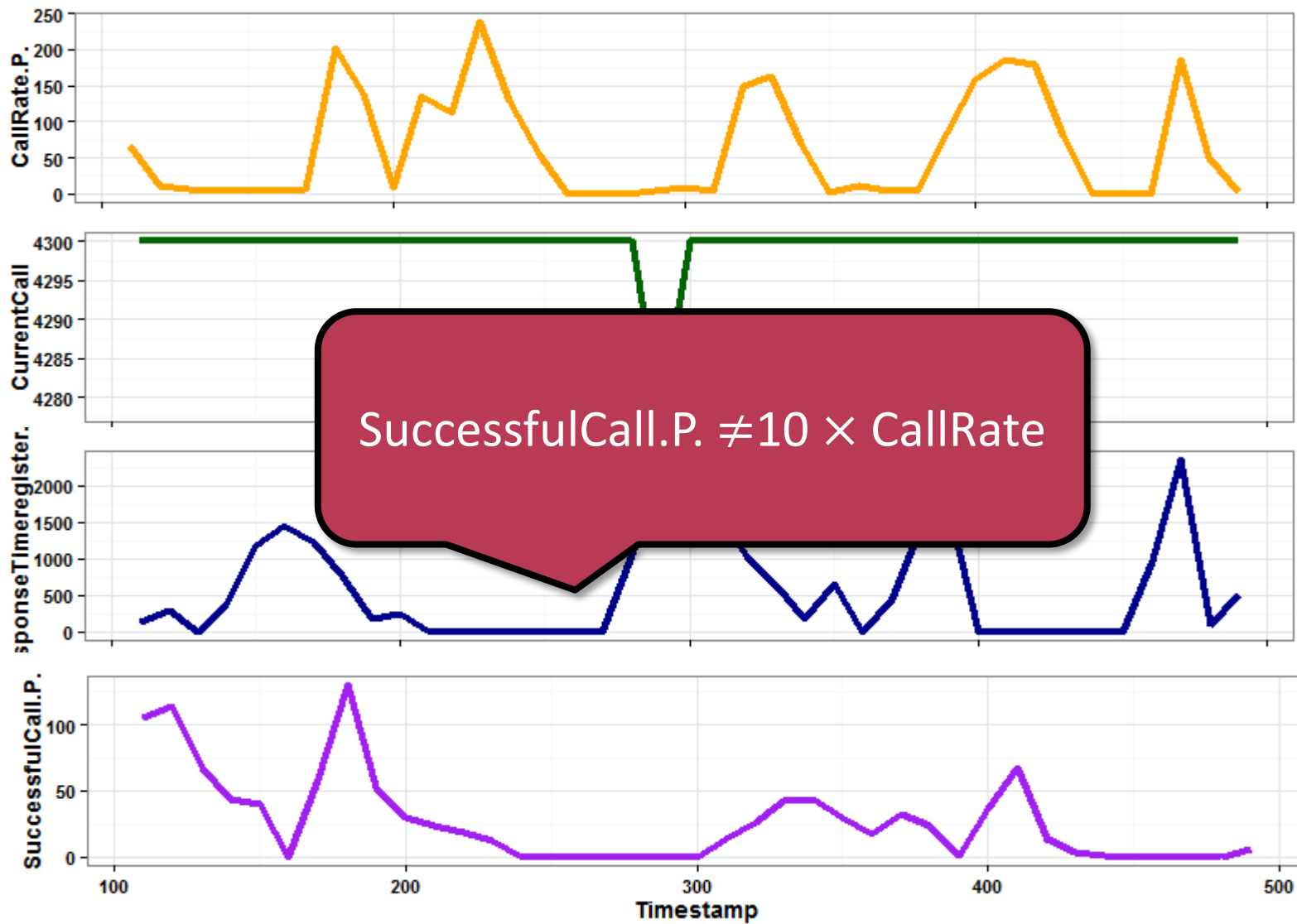
Minden kérést kiszolgáltunk, így
 $\text{SuccessfulCall.P.} \approx 10 \times \text{CallRate}$
(a mérés mintavételezése miatt)



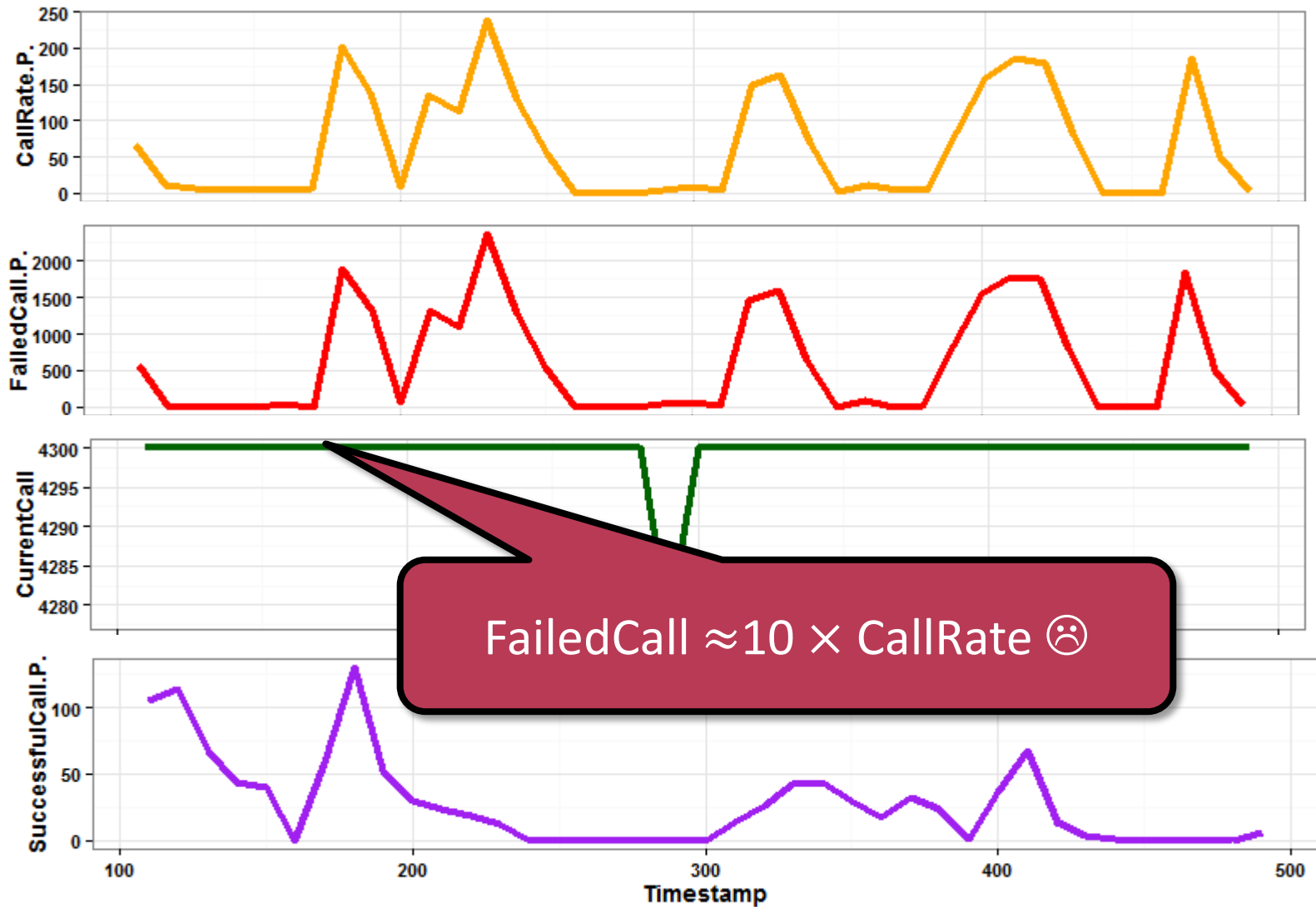
Visszacsatolás, túlterhelés



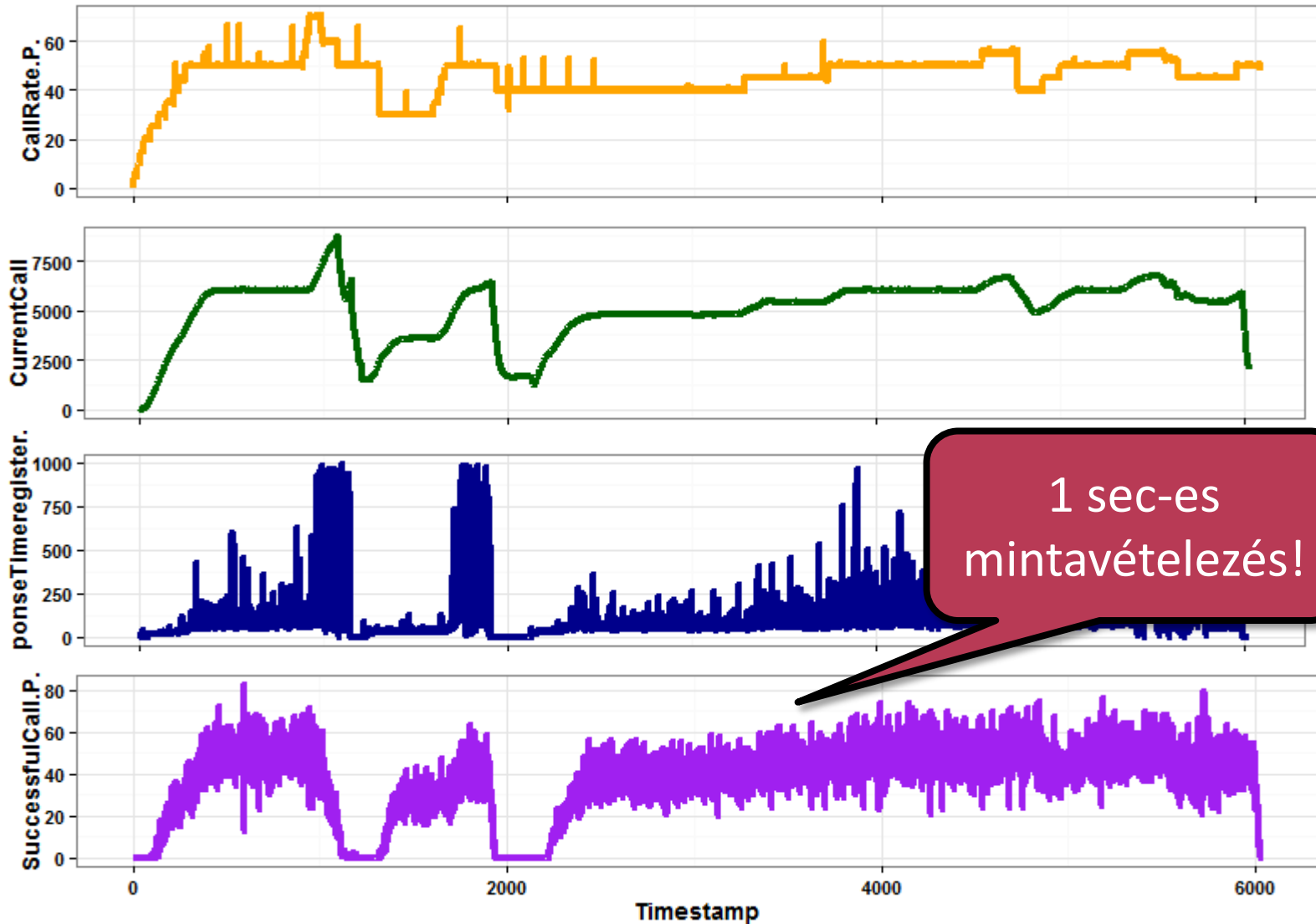
Hol sérül az állandósult állapot feltétele?



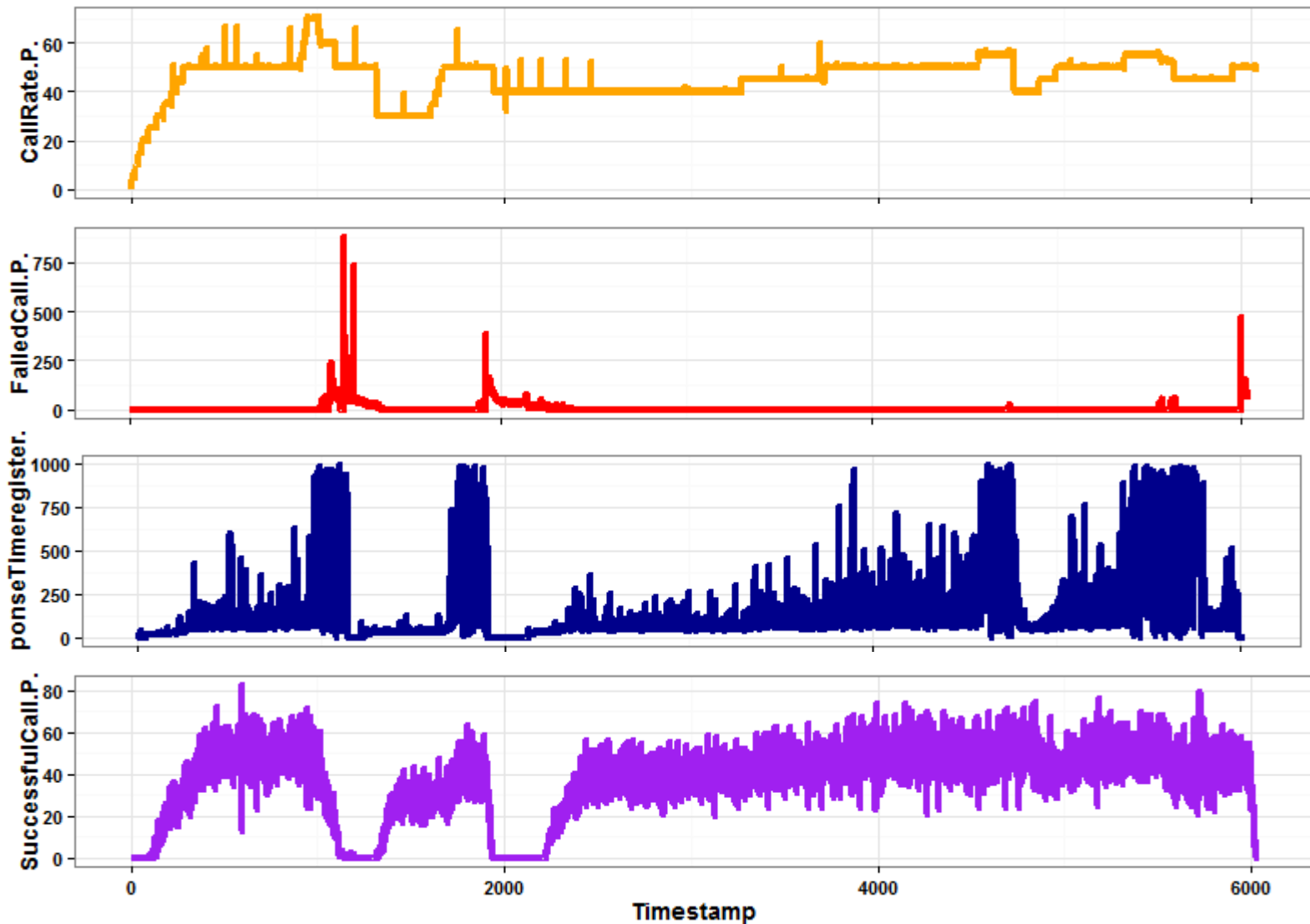
Túlterhelés: Szinte minden kérés hibás...



Nincs visszacsatolás, állandó terhelés



Finomabb mintavételezés



Hol közelítünk?

- A gyakorlatban az értékek nehezen mérhetőek
 - (pl. válaszidő ingadozik, felpörgés, ...)
- Az alkalmazások versengenek
 - ($2 * X \neq X + X$)
- Erőforrások közt választani kell
 - terheléselosztó is kritikus
 - Pl. ugyanannak a felhasználónak a kérései ugyanoda
- Konkrét beérkezési sorrendtől/mintától eltekintünk
 - Pont ez a Little-törvény előnye...
- Egy feladat végrehajtása lehet adatfüggő
- A rendszer felépítése/paraméterei változhatnak



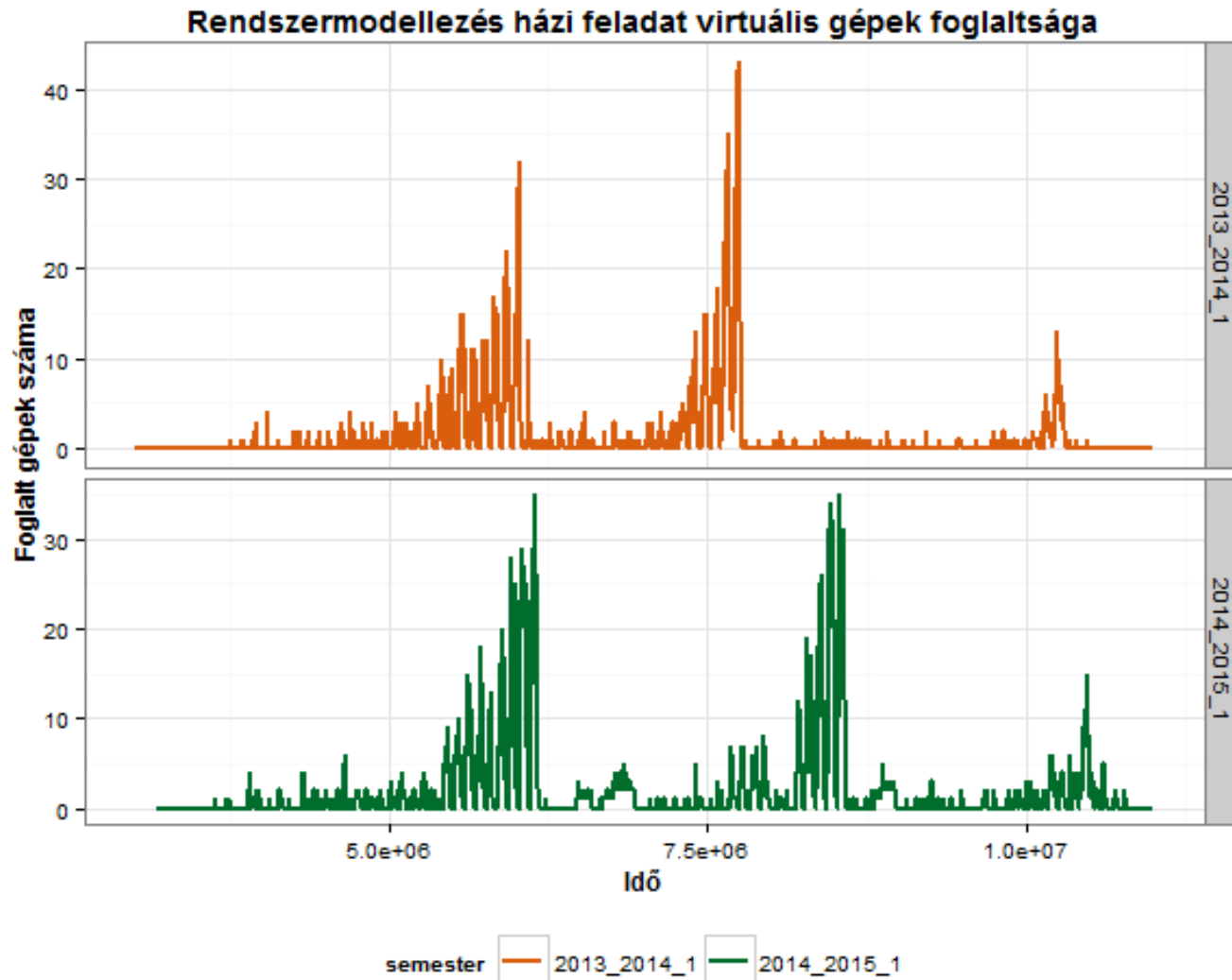
“A kezemet figyeljék, mert...”
(kép: wikipedia)

TERHELÉS VÁLTOZÁSA

Milyen jellegű a terhelés?

- Eddig:
 - Átlagos értékekkel számoltunk
 - A rendszer viselkedését a *terhelés (intenzitás)* függvényében néztük
 - De: valójában nem (feltétlenül) kiszámíthatóan nő a terhelés
- Valójában
 - A rendszer viselkedése *időben* változik
 - Ennek műszaki hatásai vannak
 - Váltás feladatok közt, erőforrásfoglalás, stb. (pl. Operációs rendszerek)

Rendszermodellezés HF a felhőben



Rendszermodellezés (7. félév) a felhőben

Rendszermodellezés házi feladat virtuális gépek foglaltsága

1. fázis: specifikáció

2. fázis: teljes feladat

Pótleadás

It már a javítás is felhőben történt...

