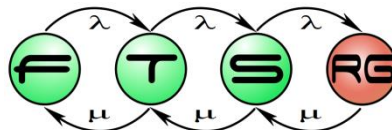


Teljesítménymodellezés

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



Remo Verseny

2019.04.30.
16:15-18:00

https://vik.hk/verseny/rendszermodellezes_2019_tavasz/

Emlékeztető

■ Egyensúlyi állapot:

- Átlagos értékekkel számolunk
- $\lambda = X$ (érkezési ráta = átbocsátás)

■ Átbocsátóképeség ($X^{\max}$):

- Az elérhető maximális átbocsátás
T átlagos feldolgozási idő mellett
- $X^{\max} = \frac{K}{T}$ (K darab erőforráspéldány esetén)

■ Kihasználtság (U):

- Az átbocsátás és az átbocsátóképeség aránya
- $U = \frac{X}{X^{\max}} = \frac{X}{\frac{K}{T}} = \frac{X}{K} \times T$ (K erőforráspéldány esetén)

A Little-törvény

A Zipf-törvény

Terhelés változása

TARTALOM

Alapfogalmak

Terhelési diagram

Erőforrásmodellezés

Folyamatmodellek elemzése

A Little-törvény

FOLYAMATMODELLEK ELEMZÉSE

Hogyan számoljuk ki egy összetett folyamat átbecsátóképeségét?

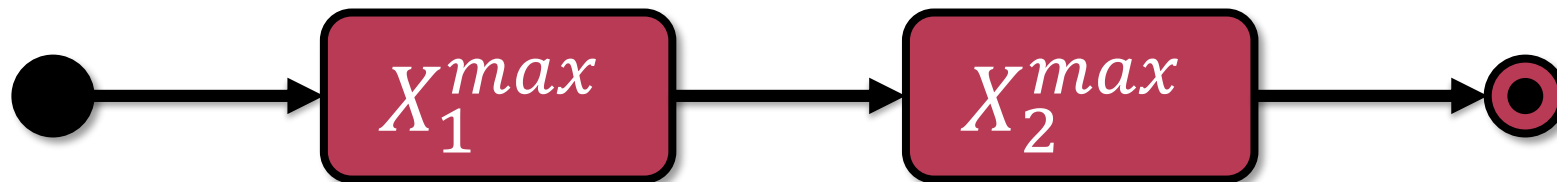
Mit tudunk tervezéskor?

- Általában tevékenységekhez rendelünk erőforrást
 - A tevékenység (átlagos) végrehajtási ideje is adott
→ X^{max} számítható a tevékenységre
- Pl. Neptunban a tárgyfelvétel az adatbázisszervert 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 átbocsátóképessége?

Szekvenciális komponálás

χ^{max}



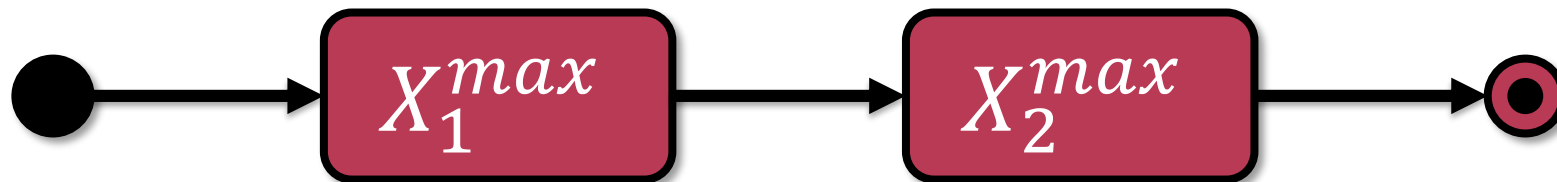
$$\chi^{max} = \min(\chi_1^{max}, \chi_2^{max})$$

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}

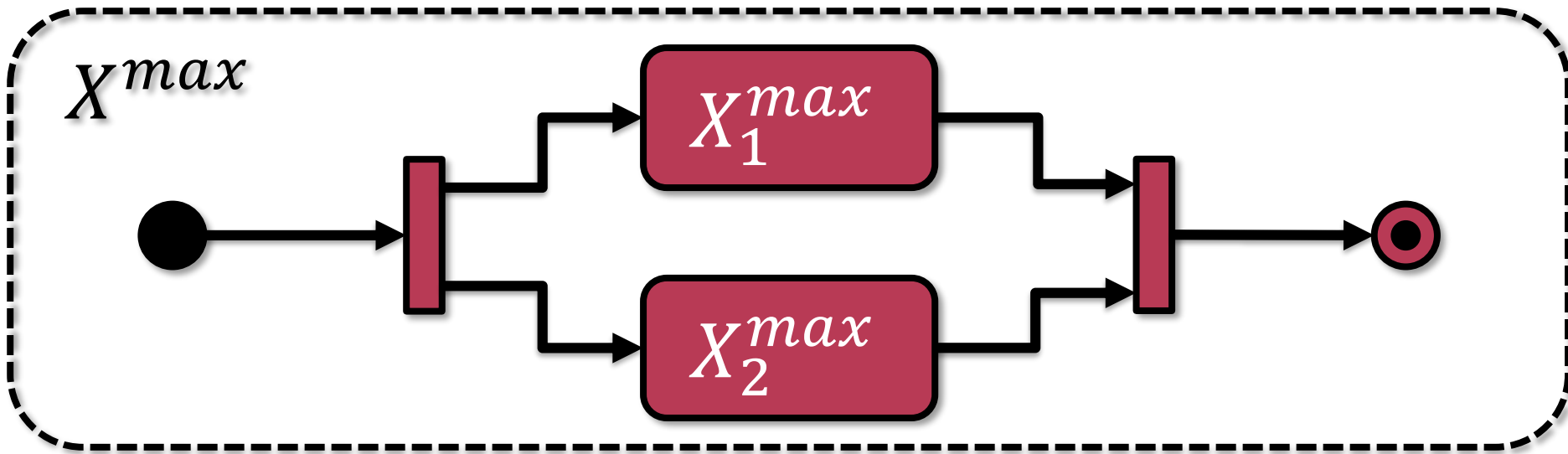


$$X^{max} = \min(X_1^{max}, X_2^{max})$$

Szűk keresztmetszet:

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

Párhuzamos komponálás

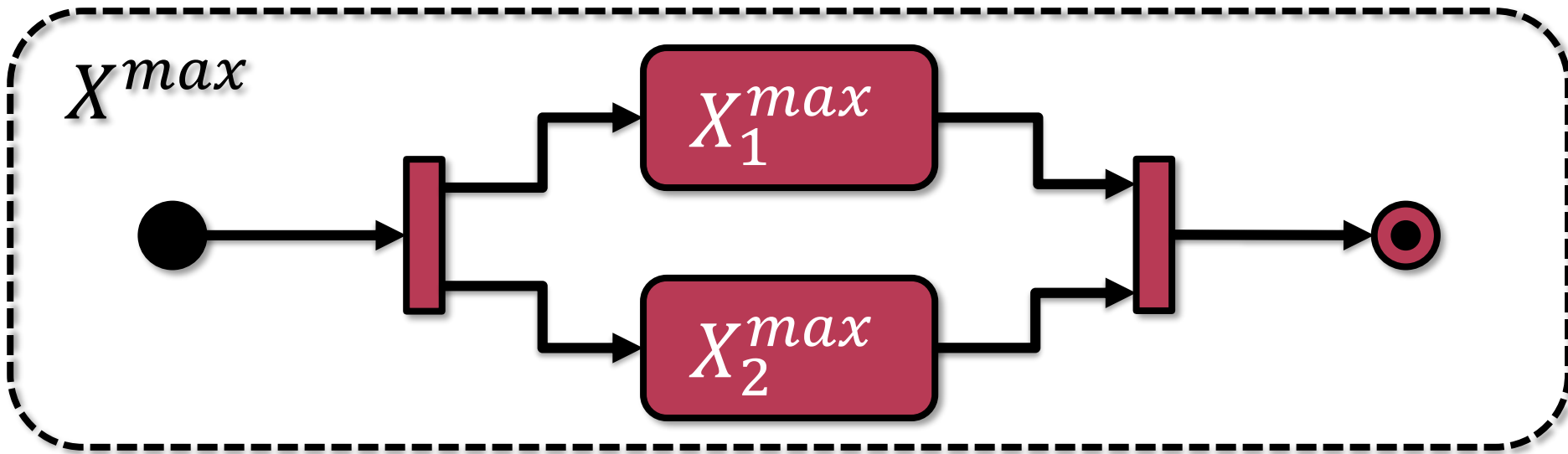


$$X^{max} = \min(X_1^{max}, X_2^{max})$$

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

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

Párhuzamos komponálás

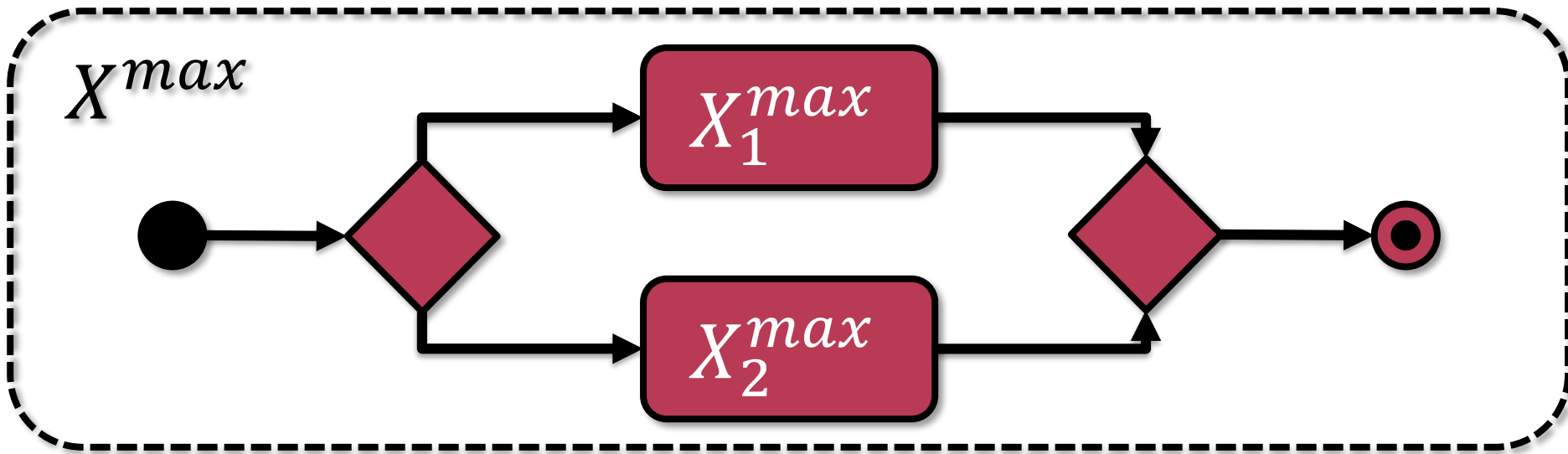


$$\chi^{max} = \min(\chi_1^{max}, \chi_2^{max})$$

Szűk keresztmetszet:

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

Komponálás szabad választással

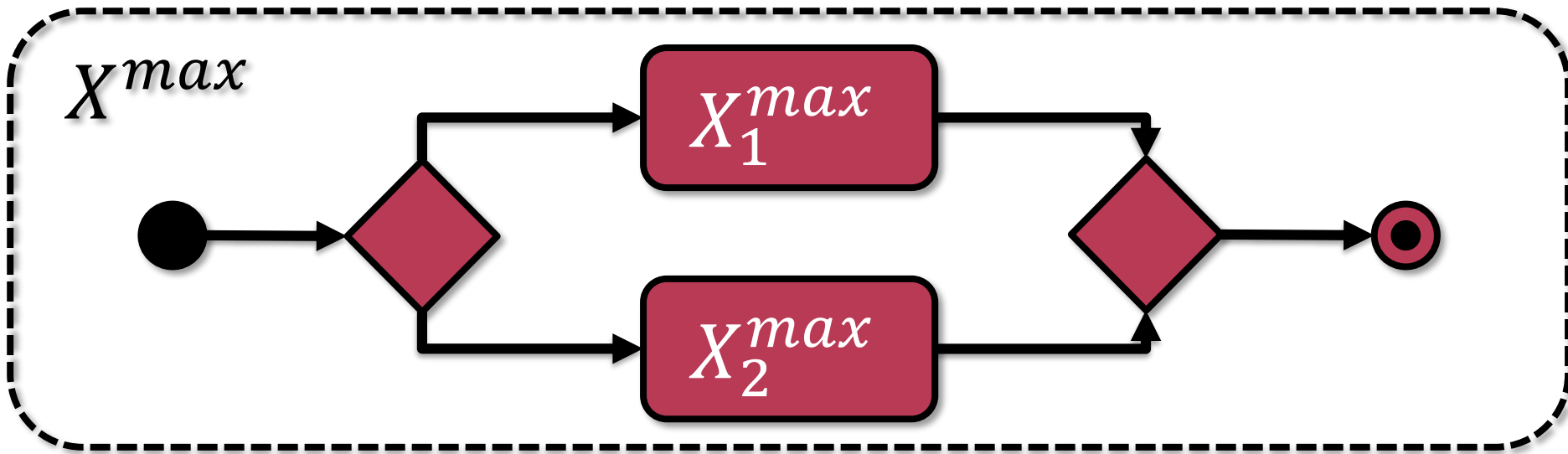


$$\chi^{max} = \chi_1^{max} + \chi_2^{max}$$

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

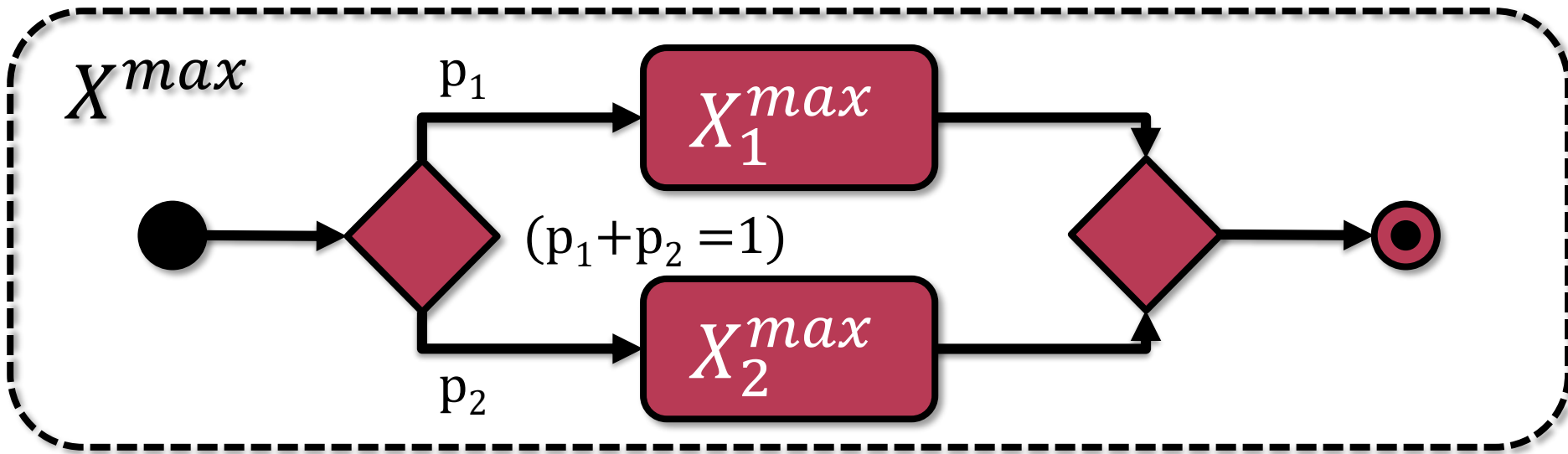


$$X^{max} = X_1^{max} + X_2^{max}$$

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

Feltétel: minden erőforráson ugyanannyi ideig tartson

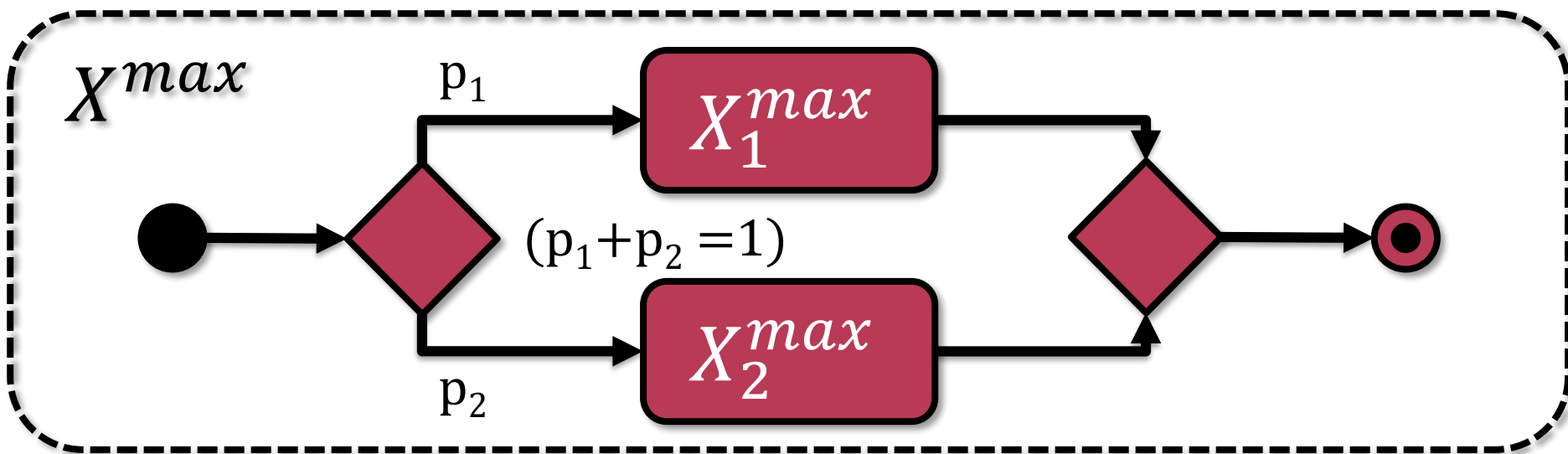
Komponálás kötött arányú választással



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

A tokenek 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.

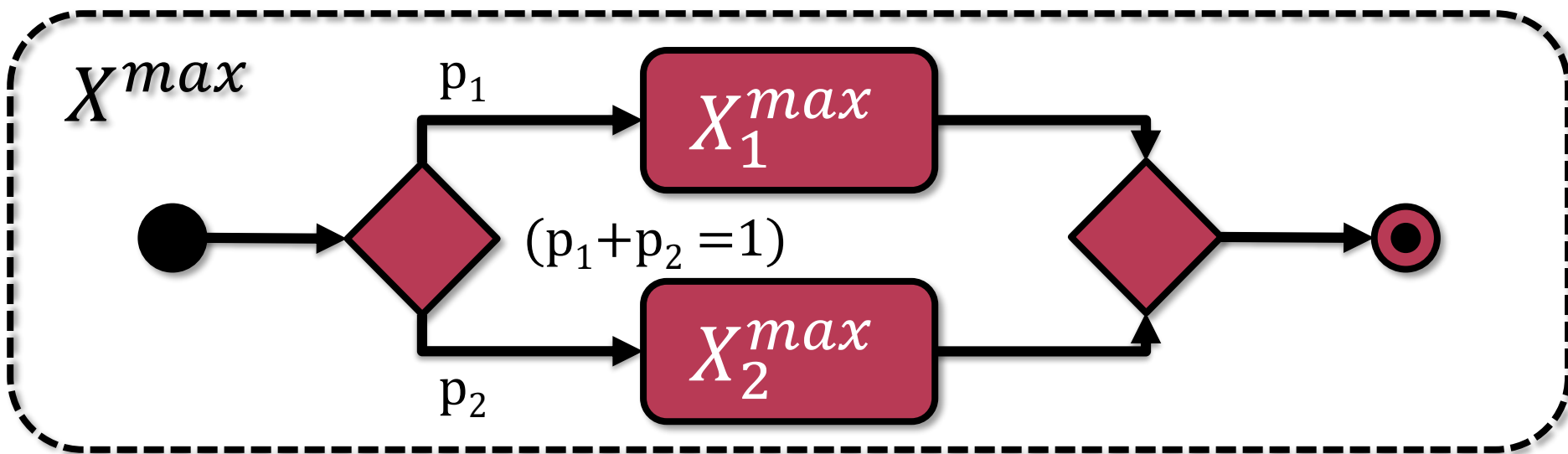
Komponálás kötött arányú választással



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

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

Komponálás kötött arányú választással



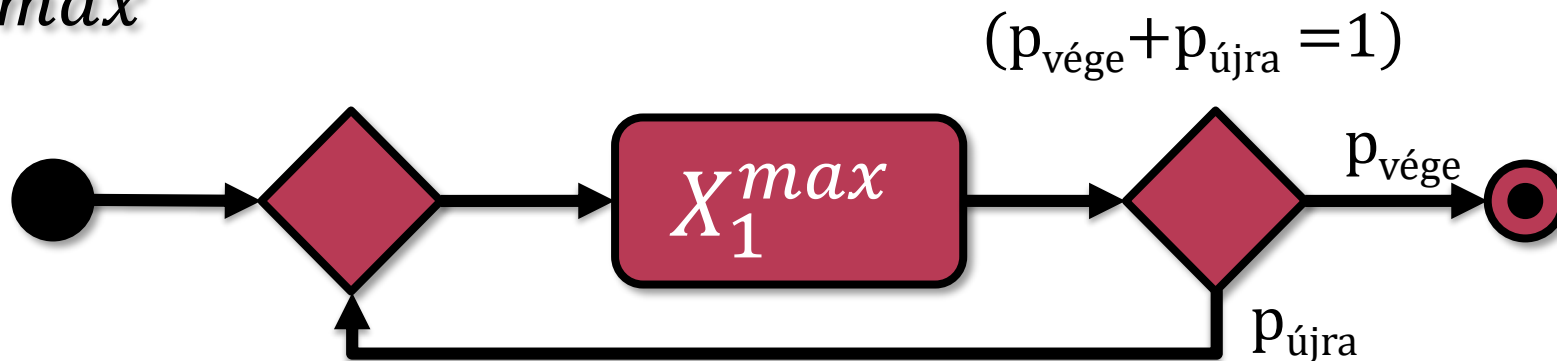
$$X^{max} = \min\left(\frac{1}{p_1} \times X_1^{max}, \frac{1}{p_2} \times X_2^{max}\right)$$

Szűk keresztmetszet:

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

Komponálás ciklussal

X^{max}

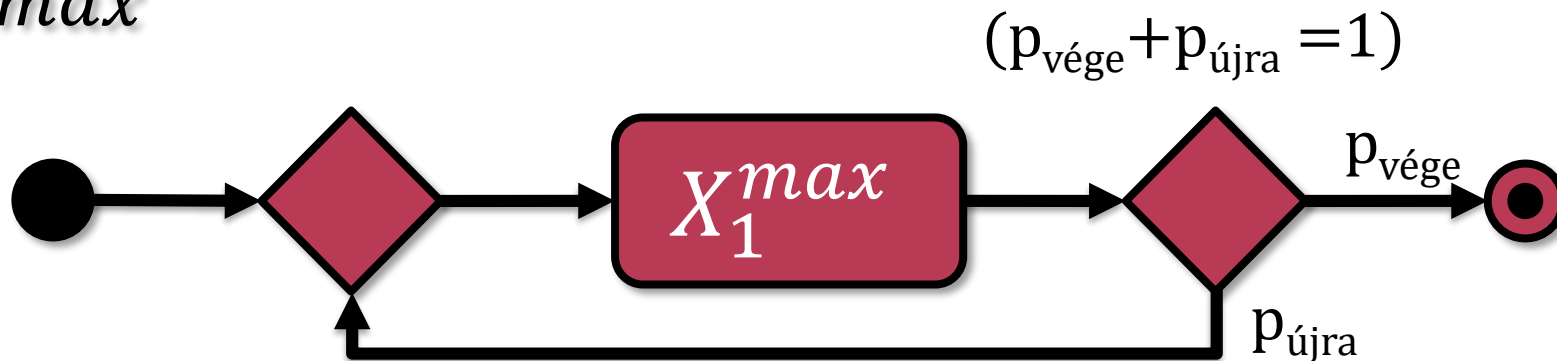


$$X^{max} = \frac{1}{p_{vége}} \times X_1^{max} = p_{vége} \times X_1^{max}$$

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

Átbocsátóképeség a vizitációs szám ismeretében:

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

Folyamat

Elemi tevékenység
(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

Vizitációs szám

Átbocsátóképeség 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

A Little-törvény

A Zipf-törvény

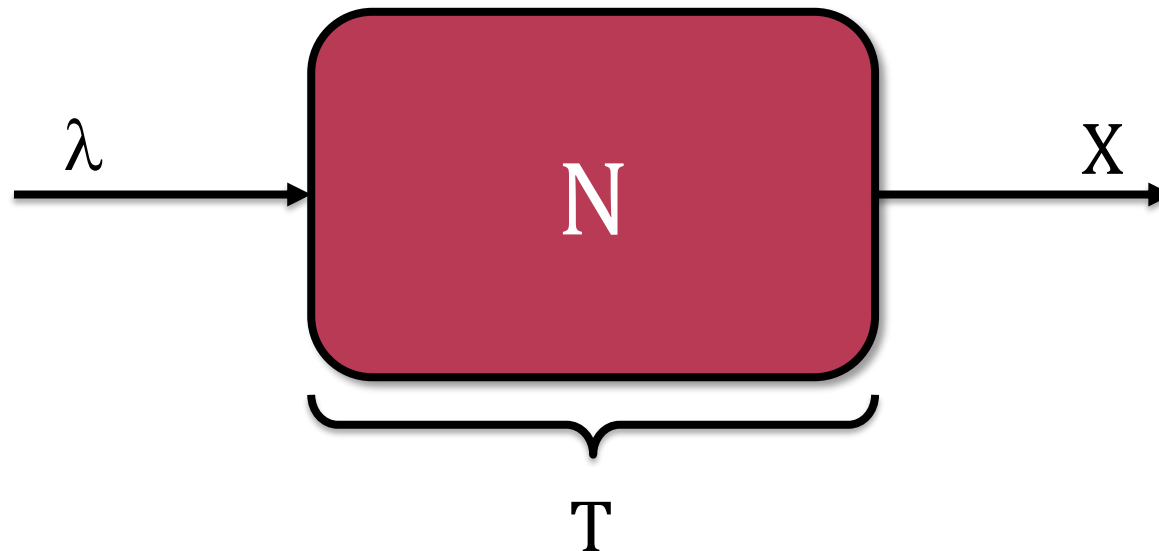
Terhelés változása

A LITTLE-TÖRVÉNY

Avagy az alapképlet

A Little-törvény

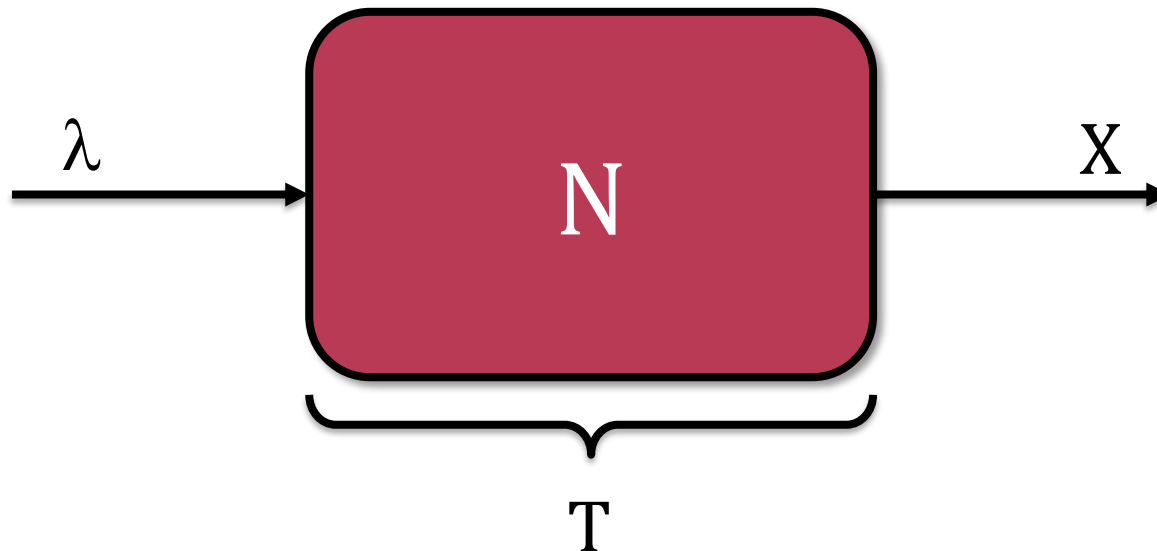
- λ : érkezési ráta $\left[\frac{darab}{s}\right]$
- X : átbocsátás $\left[\frac{darab}{s}\right]$
- T : rendszerben töltött idő $[s]$
- N : rendszerben lévő tokenek száma $[darab]$



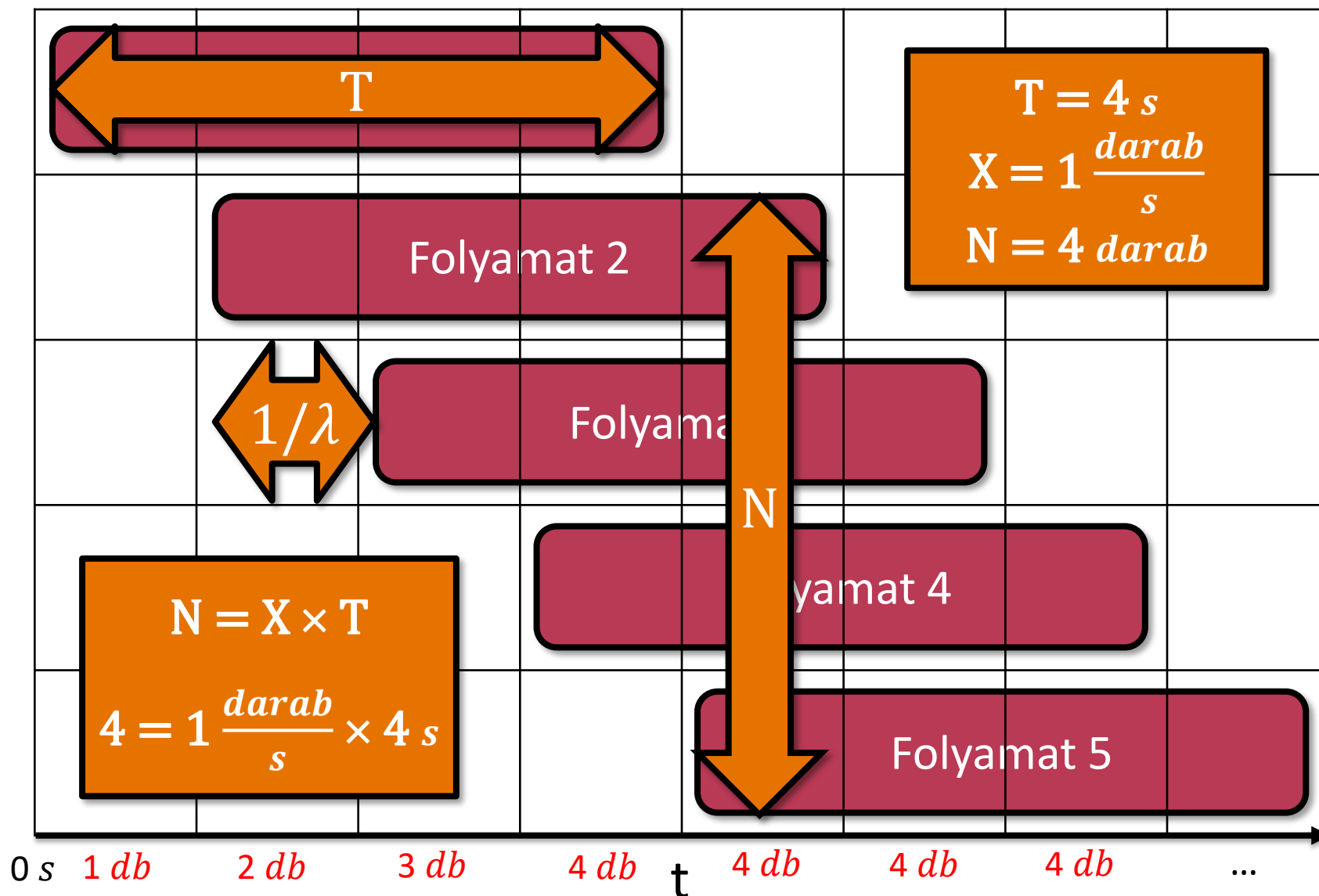
A Little-törvény

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

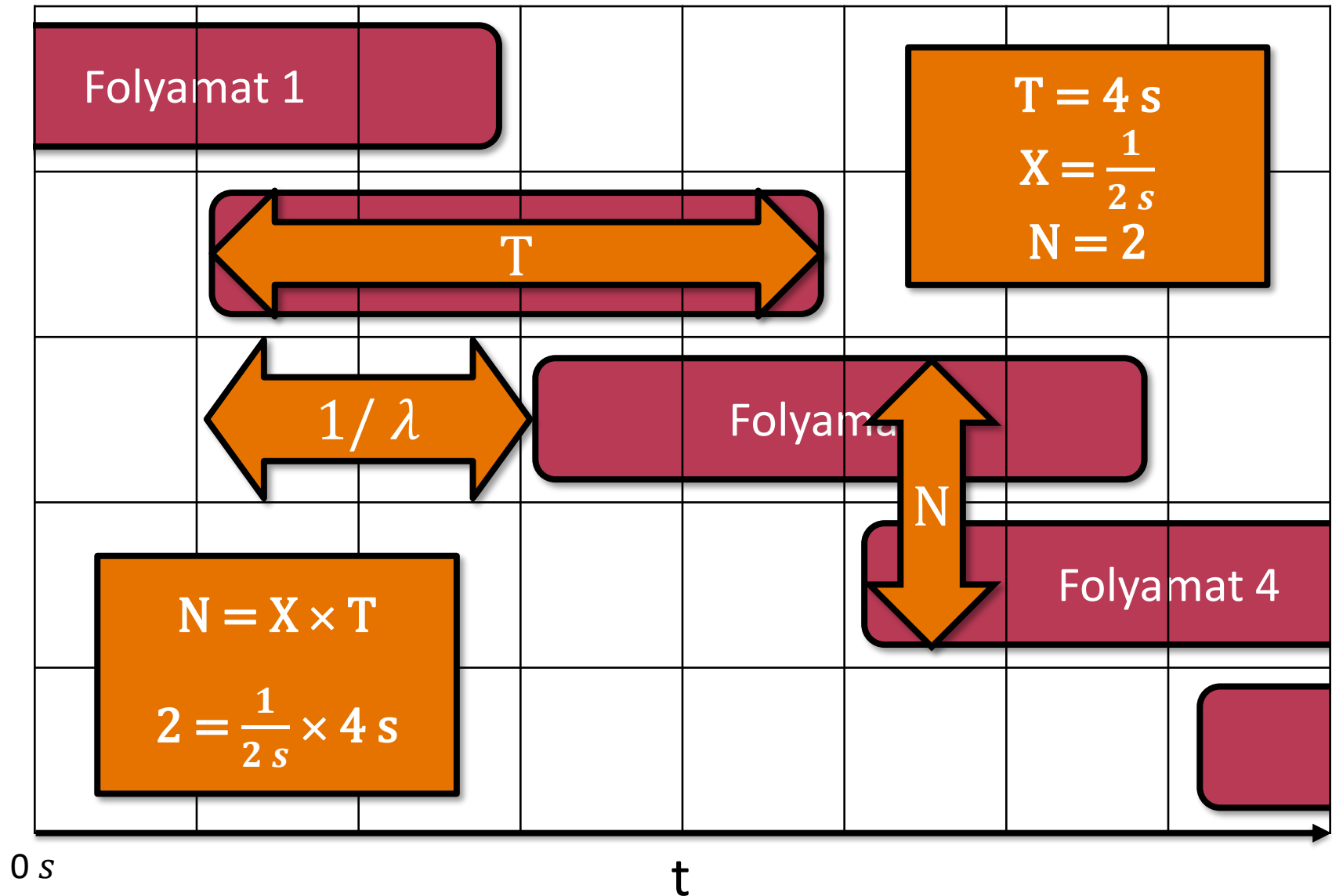
$$N = X \times T$$



A Little-törvény szemléltetése



A Little-törvény szemléltetése



Kihasználtság és a Little-törvény

- K darab erőforráspéldány:
maximum K darab kérés végrehajtás alatt
- A Little-törvény:
végrehajtás alatt álló kérések száma (N)?
- Levezethető az átlagos kihasználtság:

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

Kihasználtság K darab
erőforráspéldányra

Little-törvény
($N = X \times T$)

A Little-törvény

A Zipf-törvény

Terhelés változása

LITTLE TÖRVÉNY: GYAKORLATI PÉLDÁK

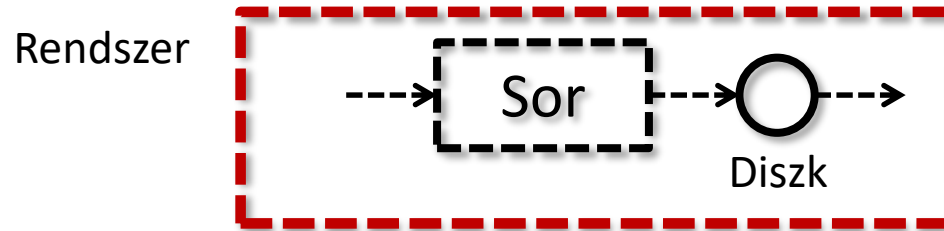
Példa



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

$$U = X \times T_{\text{diszk}} = 40 \frac{\text{kérés}}{s} \times 0,0225 \text{ s} = 0,9 = 90\%$$

Példa



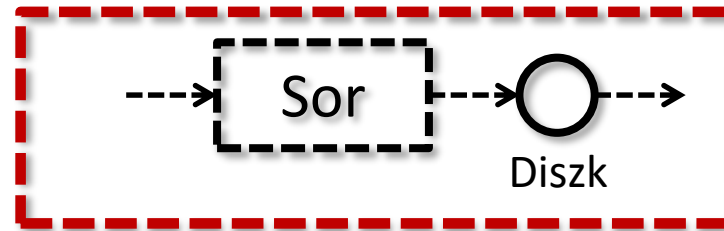
- Sorban állás is van a diszk előtt
- Diszk: 40 *kérés/s*
- Kérések átlagos száma a rendszerben: 4

Átlagos rendszerben tartózkodási idő? (T_{rendszer})

Átlagos sorban állási idő? ($T_{\text{várakozás}}$)

Példa

Rendszer



- Sorban állás is van a diszk előtt
- Diszk: 40 *kérés/s*
- Kérések átlagos száma a rendszerben

Sorbanállási és
diszk
kiszolgálási idő

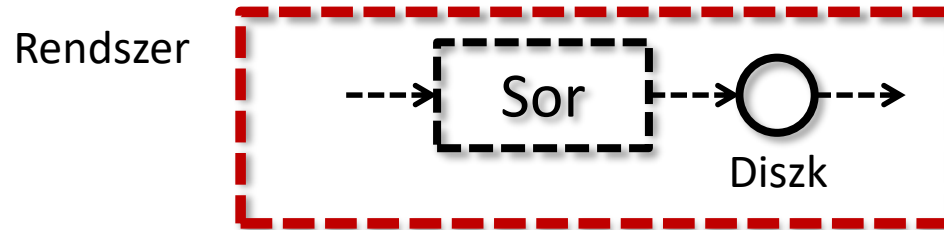
Rendszer

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

Átlagos sorban állási idő?

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

Példa



- Sorban állás is van a diszk előtt
- Diszk: 40 *kérés/s*. Átlagosan 0,9 *darab*
- Kérések átlagos száma a rendszerben: 4 *darab*

Kérések átlagos száma a sorban?

$$(N_{\text{rendszer}} - N_{\text{diszk}})$$

$$4 \text{ darab} - 0,9 \text{ darab} = 3,1 \text{ darab}$$

Little törvény a gyakorlatban

■ Szimuláció

- Dobson&Shumsky
- <https://youtu.be/UjzXQPGBaNA>

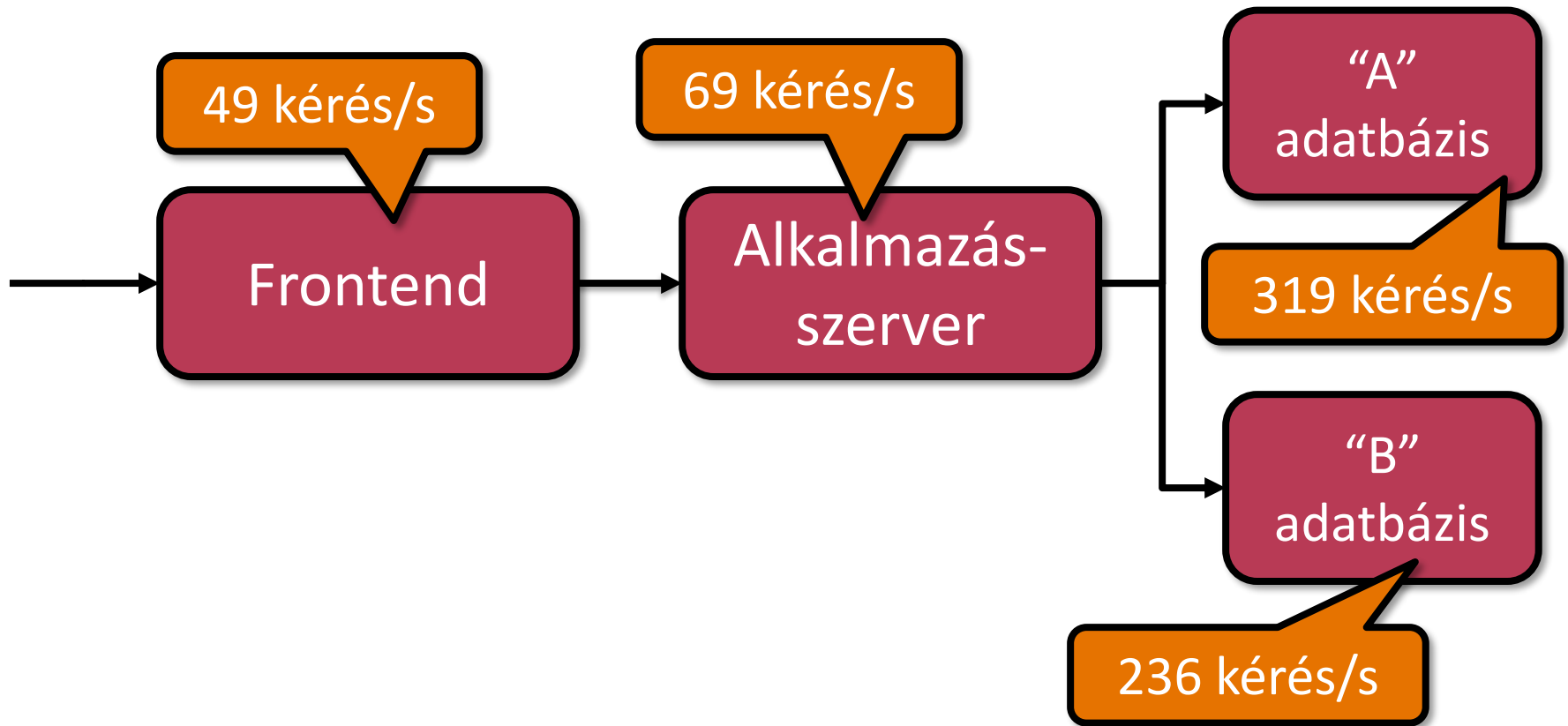
■ Miért oktatják(juk)

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

■ Példák

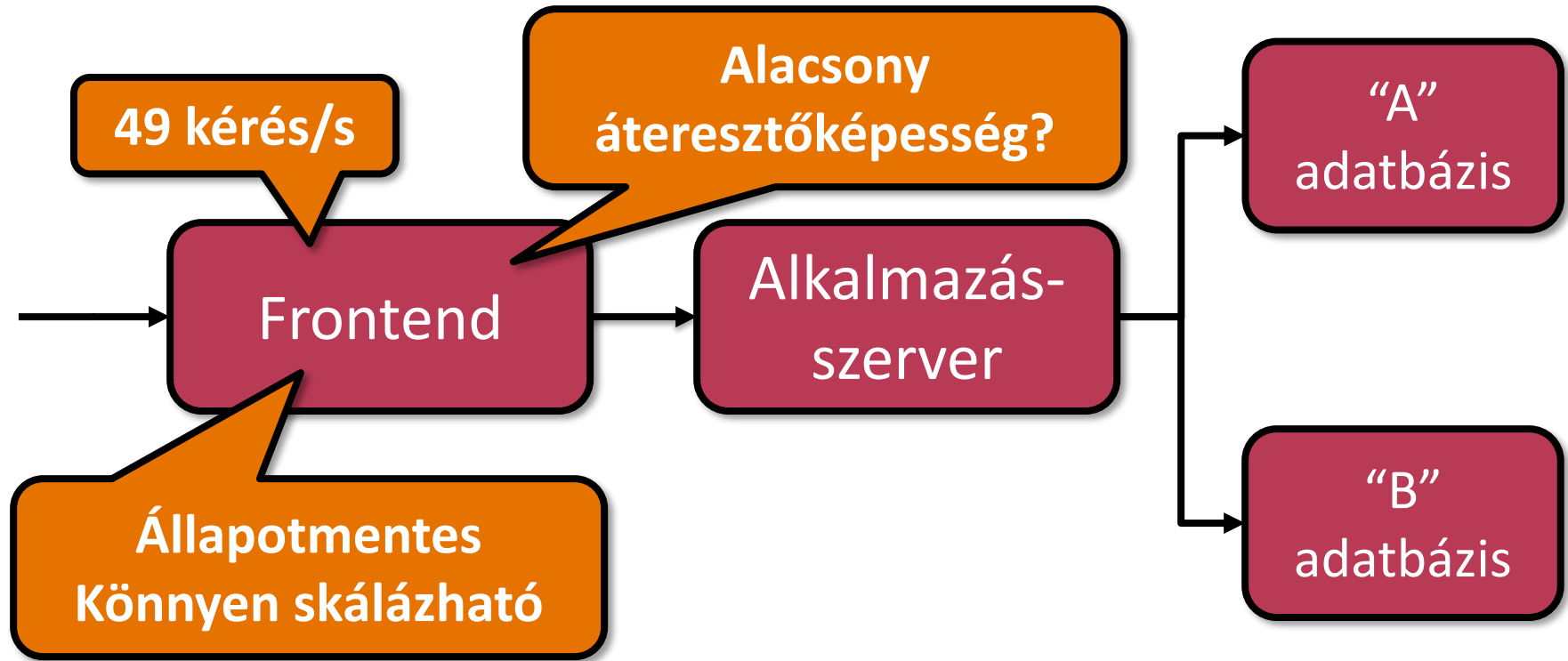
- <http://web.mit.edu/sgraves/www/papers/Little's%20Law-Published.pdf>

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



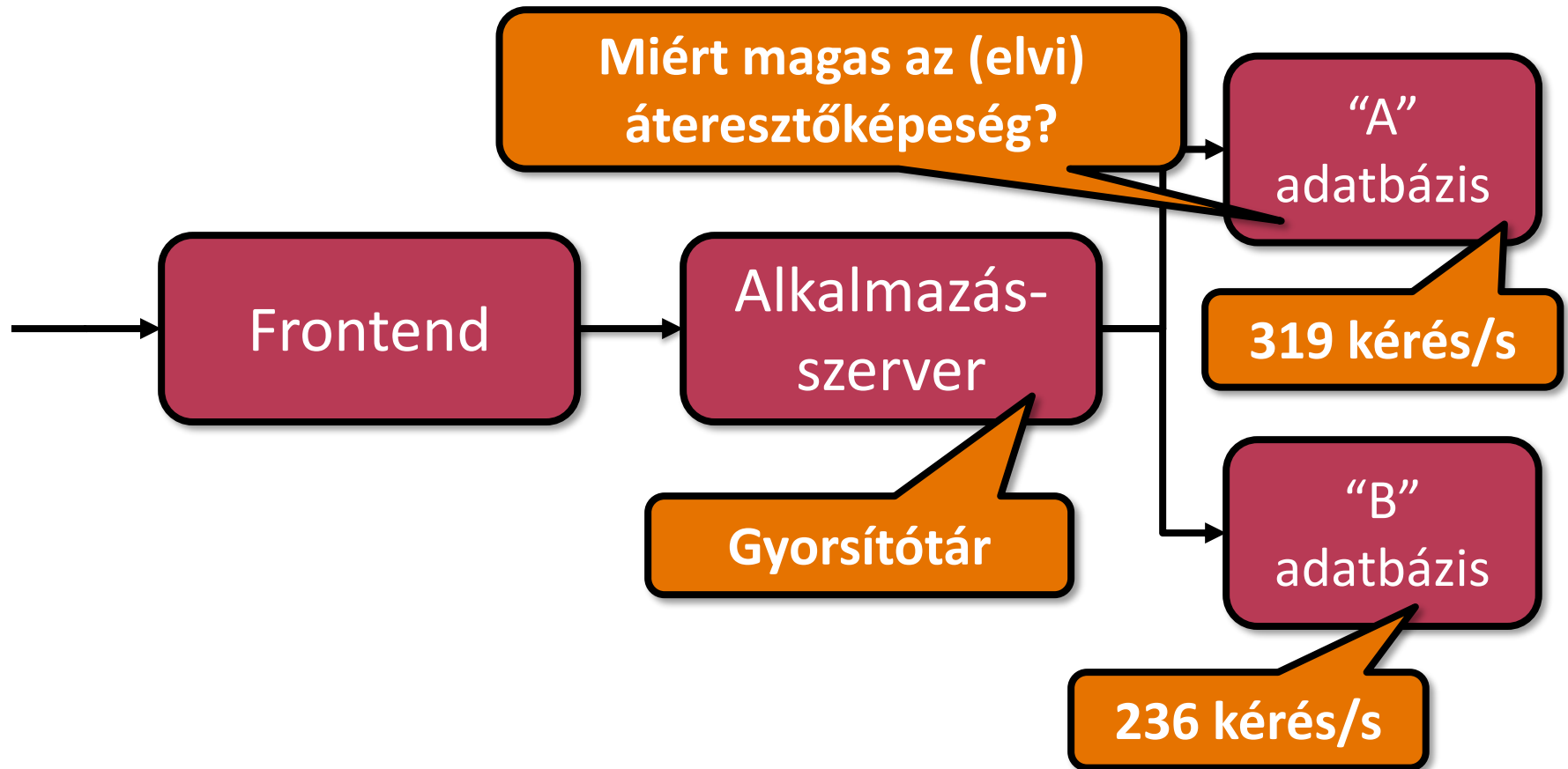
A mérőszámok itt az egész rendszerre érkező terhelésre utalnak! Például az „A. adatbázis” akkor válik szűk keresztmetszetté, ha a rendszerbe 319 lekérdezés érkezik másodpercenként.

Teljesítmény 3 rétegű architektúrában



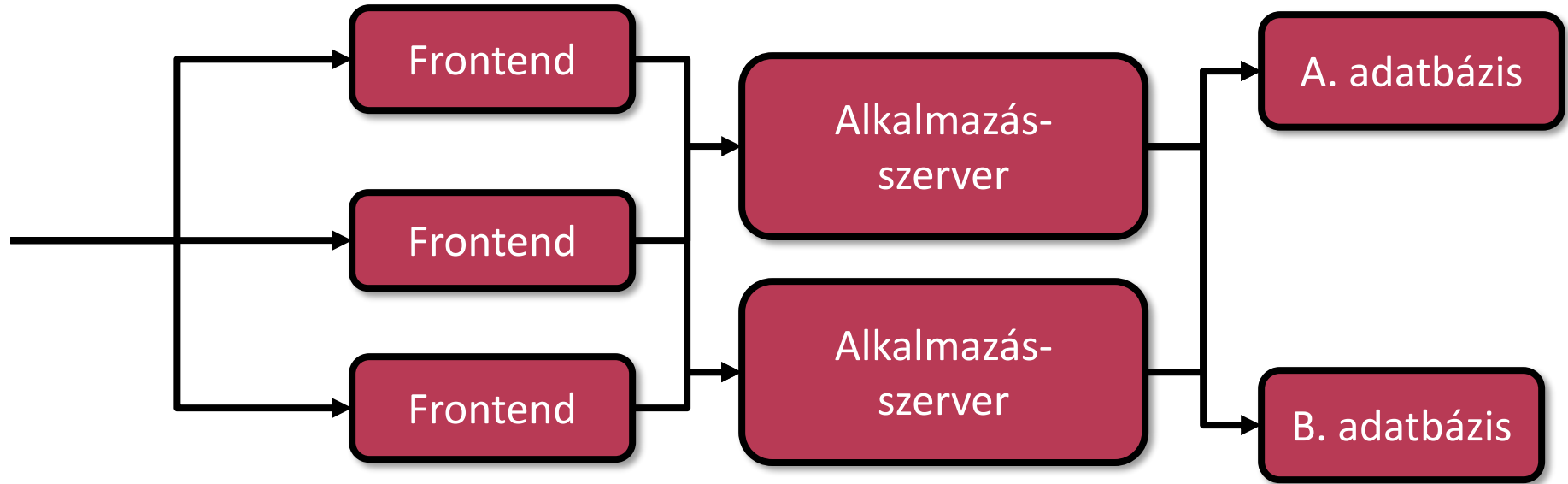
A mérőszámok itt az egész rendszerre érkező terhelésre utalnak! Például az „A. adatbázis” akkor válik szűk keresztmetszetté, ha a rendszerbe 319 lekérdezés érkezik másodpercenként.

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:)

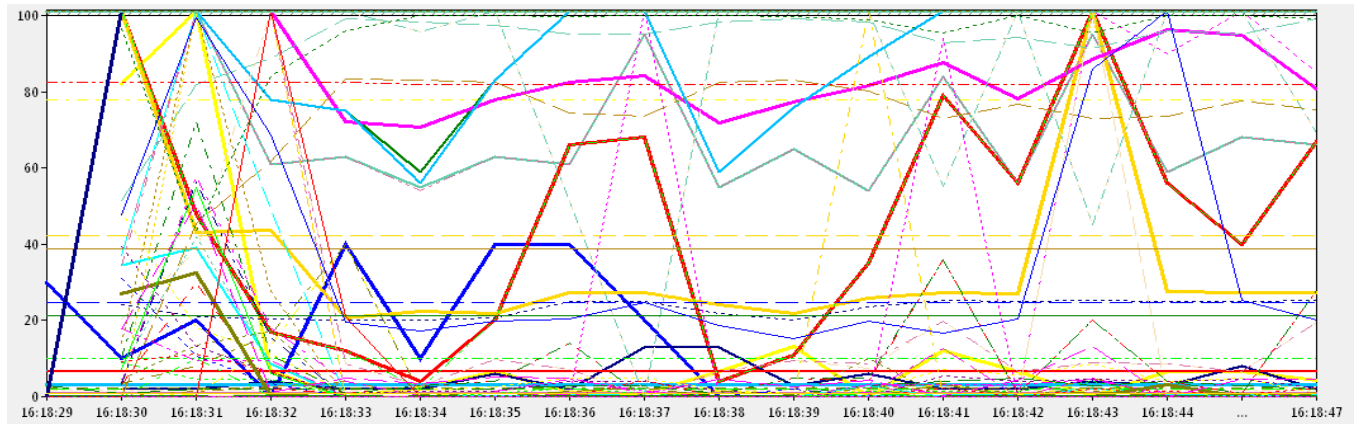
<http://www.projectclearwater.org/wp-content/uploads/2013/05/Clearwater-Deployment-Sizing-10-Apr-13.xlsx>

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

Mit mérjünk/mi a lényeges?

- Metrikák “kicsiben”

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

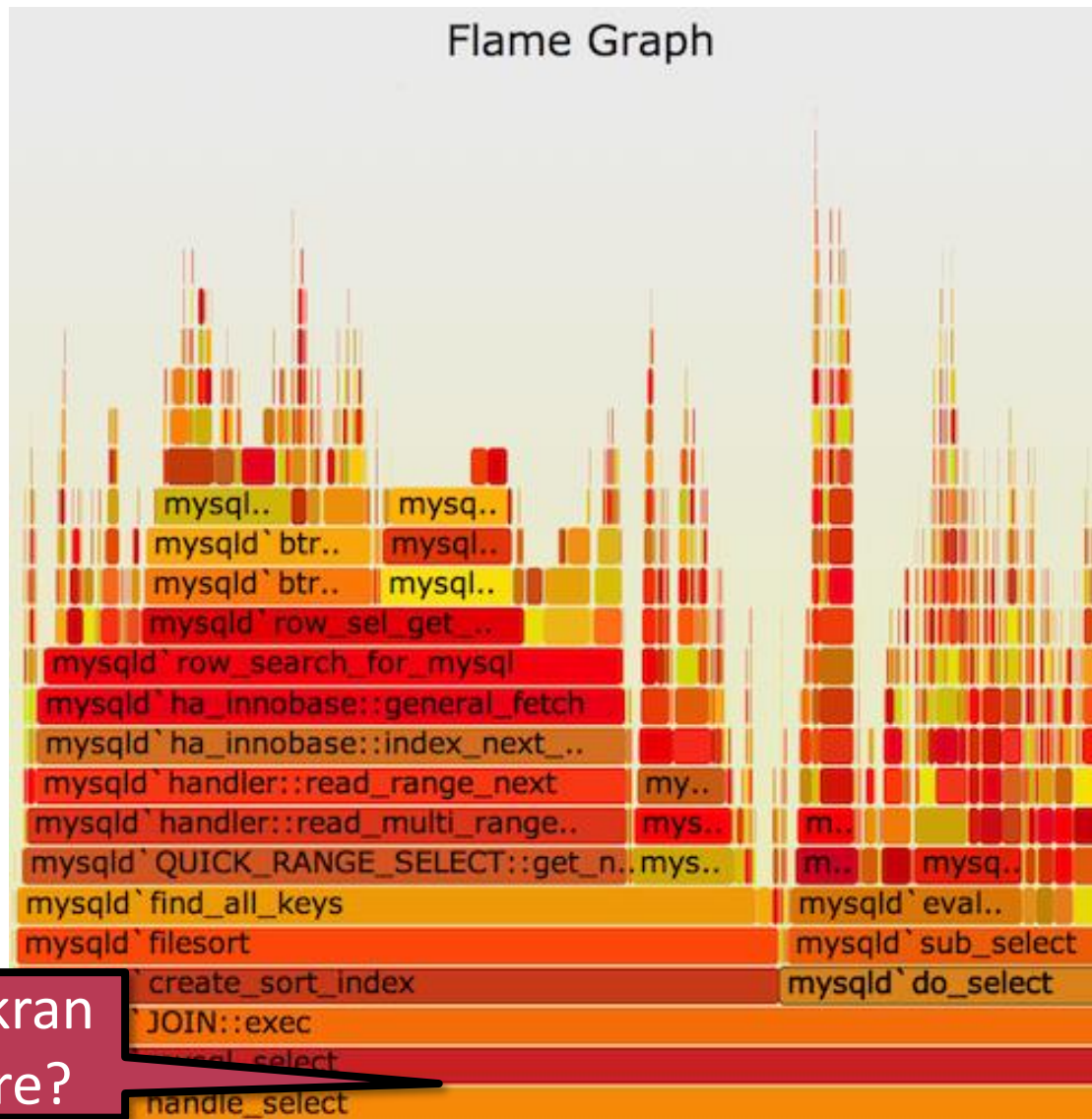


- Metrikák “nagyban”

- Pl. virtualizált rendszer

- Melyik az érdekes?

Példa: mit számol ez a gép enyit?

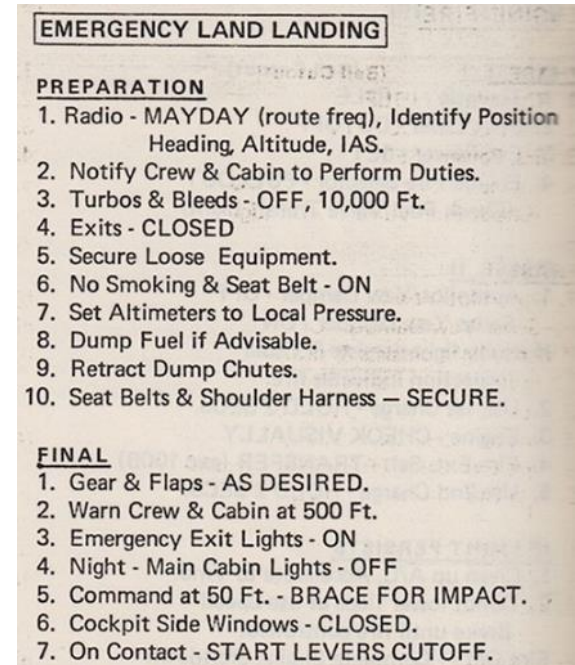


Mi milyen gyakran
hajtódott végre?

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

USE módszer

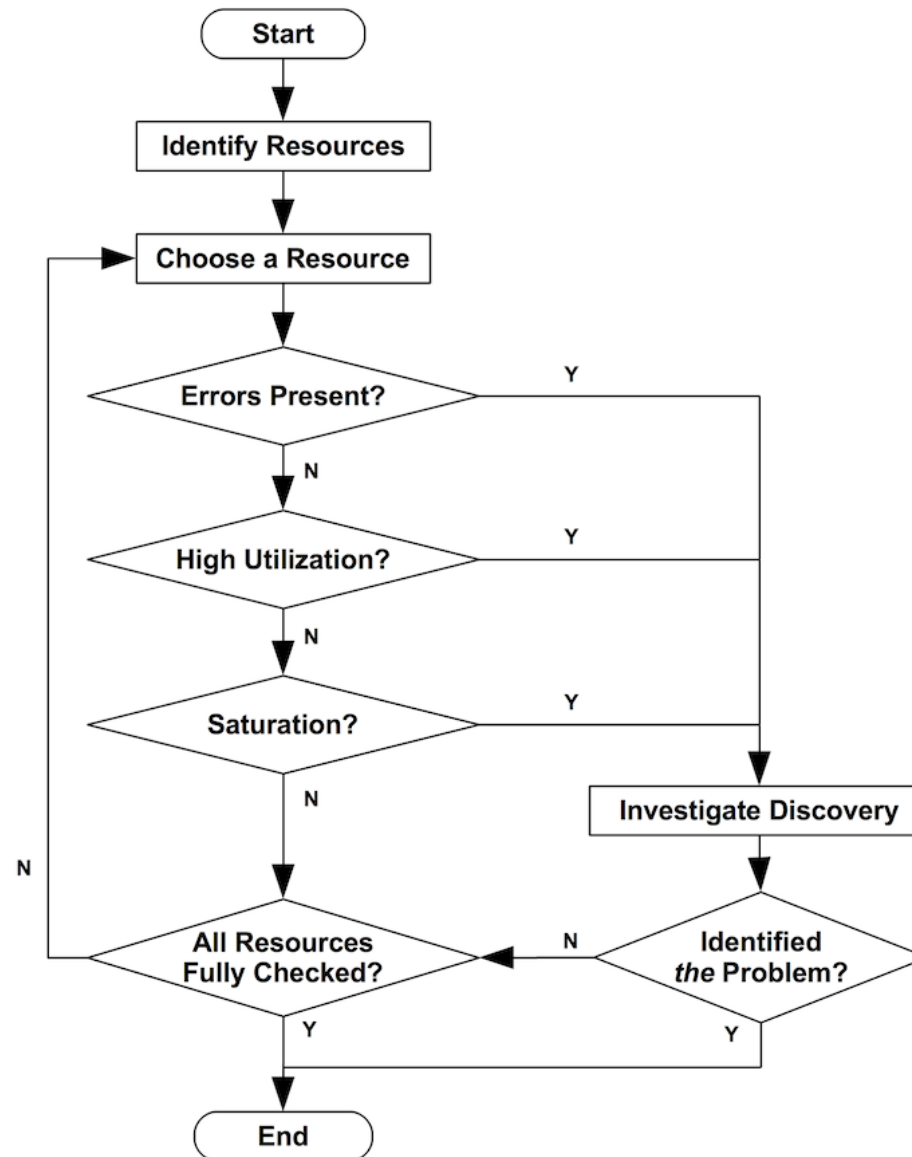
- Utilization (kihasználtság)
- Saturation (telített állapot)
- Errors (hibák)
- Példa erőforrások
 - HW: CPU, Memória, I/O, hálózat, csatolók...
 - SW: szálak száma, zárak, folyamatok száma, engedélyezett fájlok száma, ...



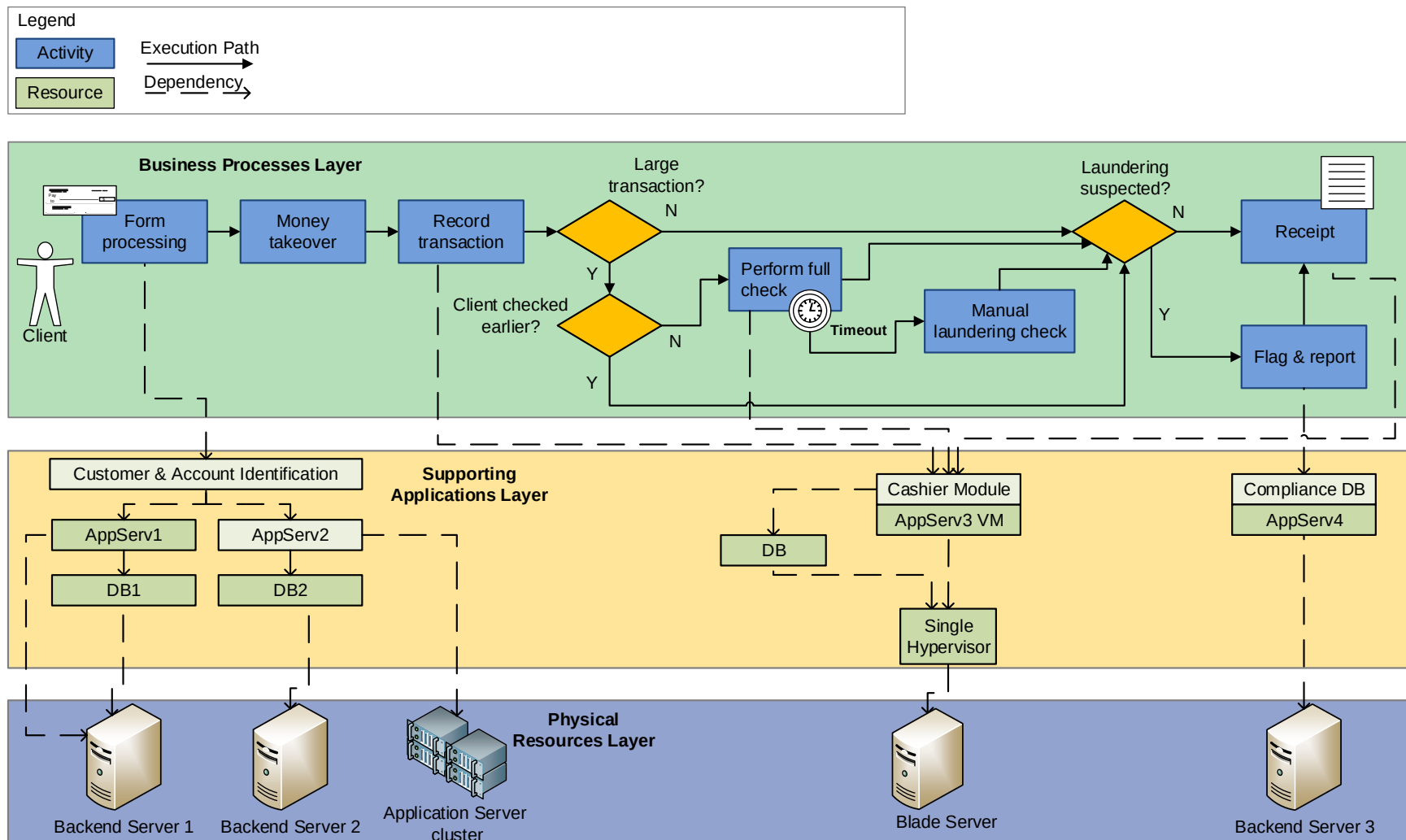
Boeing 707 ellenőrző lista

<http://www.brendangregg.com/usemethod.html>
<http://www.brendangregg.com/sysperfbook.html>

Hogyan érdemes módszeresen csinálni?



Példa: erőforrások és folyamatok együtt



Urbanics, G., Gönczy, L., Urbán, B., Hartwig, J., & Kocsis, I. (2014, October).

Combined Error Propagation Analysis and Runtime Event Detection in Process-Driven Systems.

In *International Workshop on Software Engineering for Resilient Systems* (pp. 169-183). Springer International Publishing.

Hol közelítünk?

- A gyakorlatban az értékek nehezen mérhetőek
 - (pl. válaszidő ingadozik, felpörgés, ...)
- Az alkalmazások versengenek
 - $(2 * \lambda \neq \lambda + \lambda)$
- 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



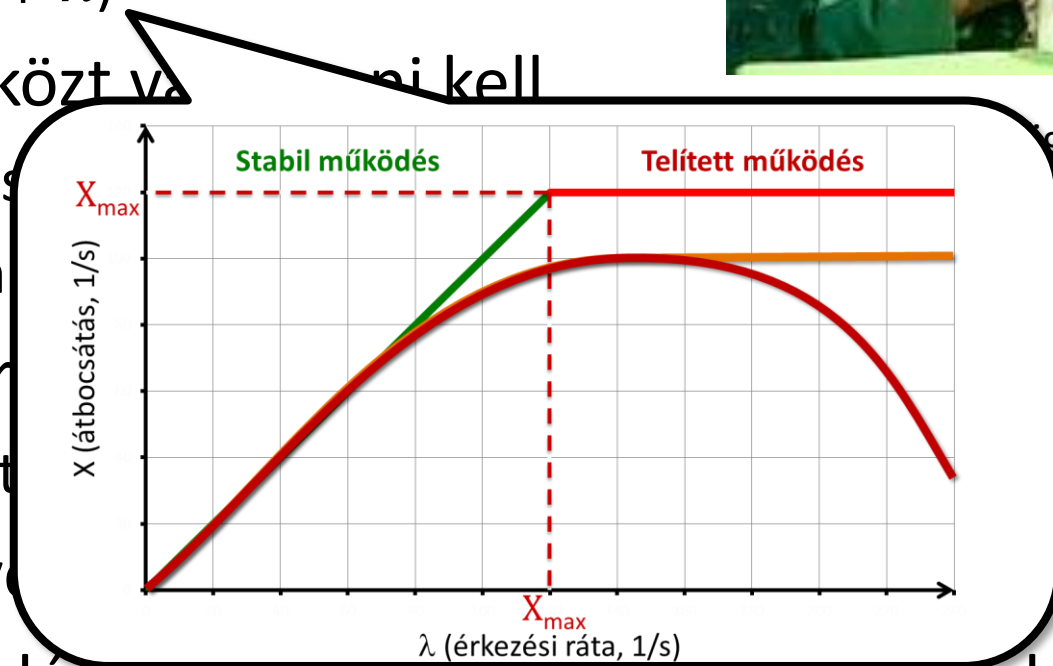
“A kezemet figyeljék, mert...”
(kép: wikipedia)

Hol közelítünk?

- A gyakorlatban az értékek nehezen mérhetőek
 - (pl. válaszidő ingadozik, felpörgés, ...)
- Az alkalmazások versengenek
 - ($2 * \lambda \neq \lambda + \lambda$)
- Erőforrások közt van versengés, ami kell



- terheléseloszlás
- Pl. ugyanannyi feladat
- Konkrét beérkezési ráta
- Pont ez a Little's law
- Egy feladat várakozási ideje
- A rendszer felépítése/parameterai változhatnak



A Little-törvény

A Zipf-törvény

Terhelés változása

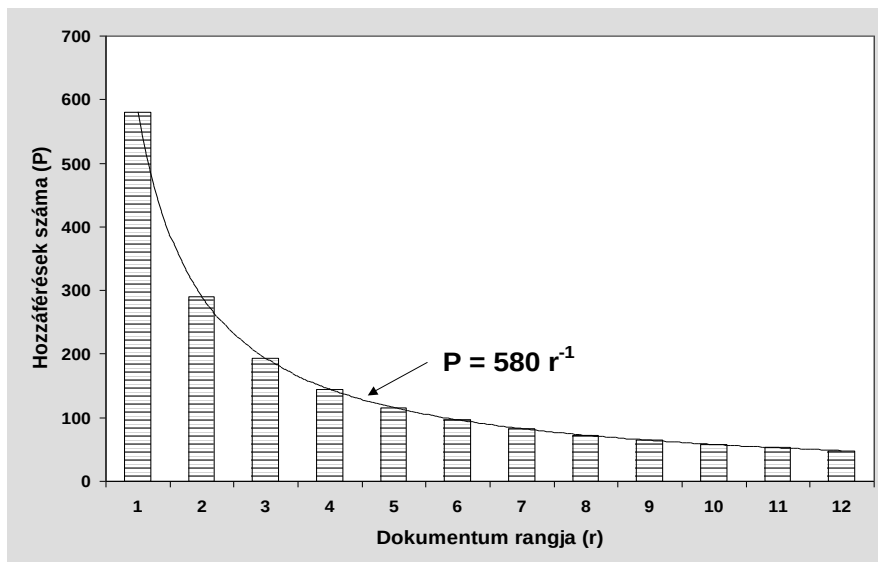
TERHELÉSMODELLEK: ZIPF TÖRVÉNY

Mi a kérések tartalma?

- [illegible]

Zipf törvénye

- Eredetileg: *korpuszokban* előforduló szavak népszerűségi rangsora és előfordulási gyakorisága jellegzetes eloszlása
 - Nemcsak nyelvi szövegekre igaz



George Kingsley Zipf
(1902–1950)
amerikai nyelvész,
filológus

Zipf törvénye - Példák

- Slágerlisták
- Városok populációja rangsoruk szerint
- Internetes forgalom karakterisztikája
- Weboldalak aloldalainak népszerűsége
- Nyílt forrású rendszerek evolúciója

Zipf törvénye - Képlet

$$R_i \sim \frac{1}{i^\alpha} \qquad f \sim \frac{1}{p}$$

- R_i – az i . szó előfordulási gyakorisága
- α – a korpuszra jellemző 1 közeli érték
- Egyszerűsítve ($\alpha = 1$):
 - f (frequency) gyakoriság
 - p (popularity): a szöveg „rangja” (csökkenő sorrendben)

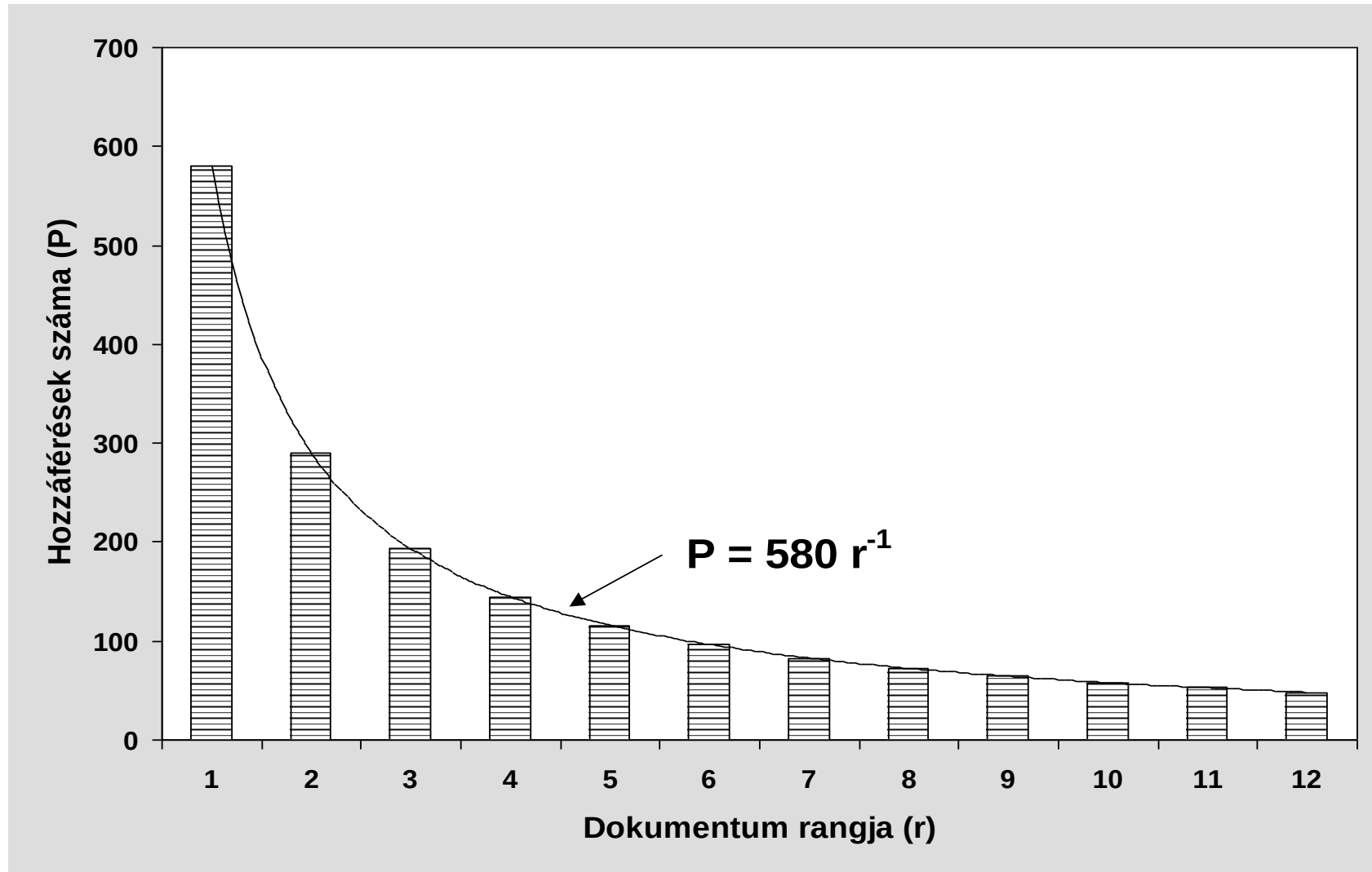
Zipf törvénye – pl. Web dokumentumokra

$$P = \frac{k}{r}$$

- P – hivatkozások (elérések)
- r – rang (1 = leggyakoribb)
- k – pozitív konstans

Bővebben: <http://www.hpl.hp.com/research/idl/papers/ranking/adamicglottometrics.pdf>

Zipf – Példa (1)

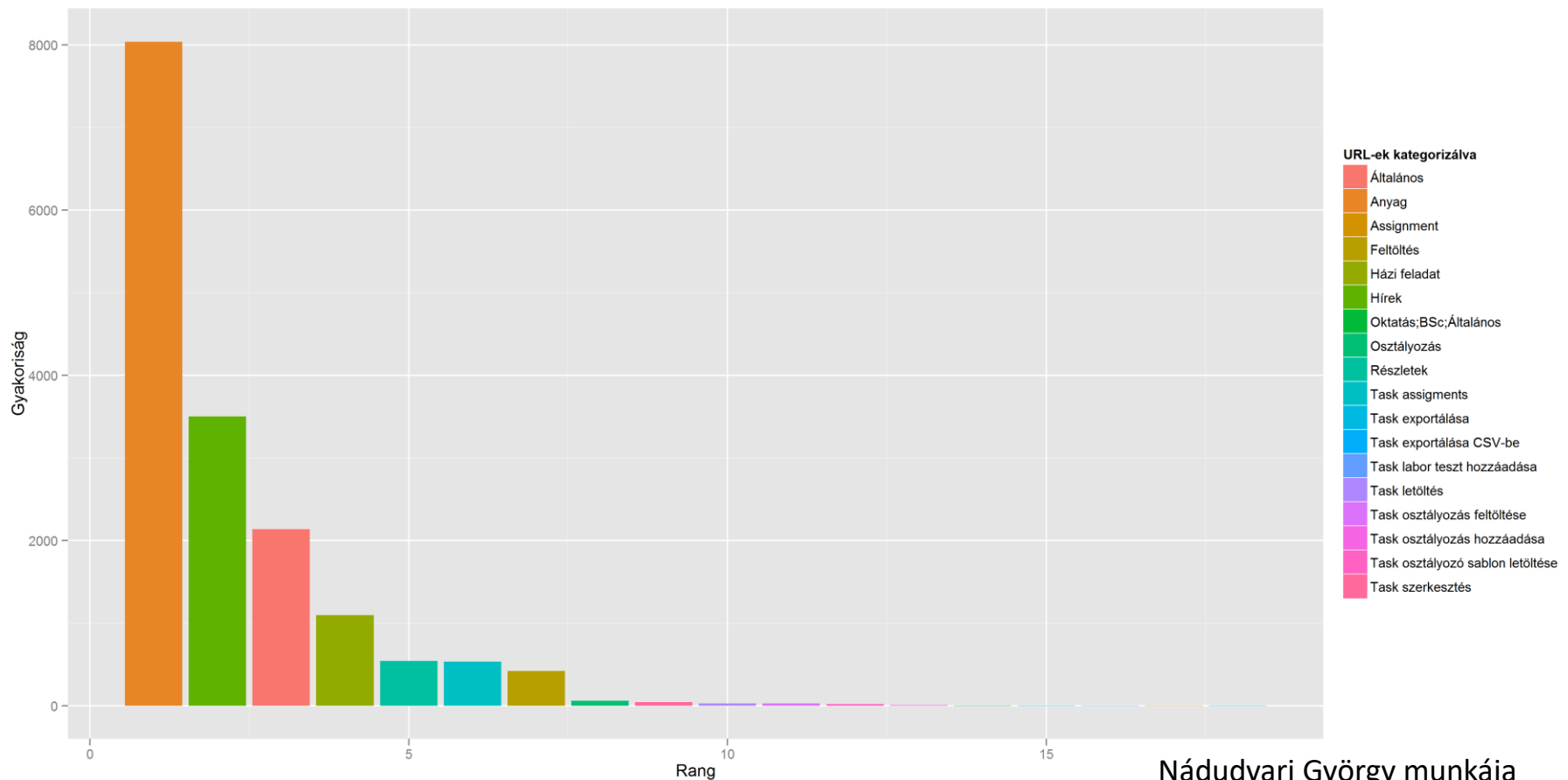


Zipf – Példa: Tanszéki honlap (1)



Zipf – Példa: Tanszéki honlap (2)

- A Rendszermodellezés tárgy oldalainak látogatottsága



Nádudvari György munkája