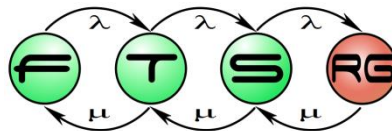


Teljesítménymodellezés

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



Emlékeztető

■ Egyensúlyi állapot:

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

■ Átbocsátóképesség ($X_{\max}$):

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

■ Kihasználtság (U):

- Az átbocsátás és az átbocsátóképesség 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 átbecsátóképeességét?

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}



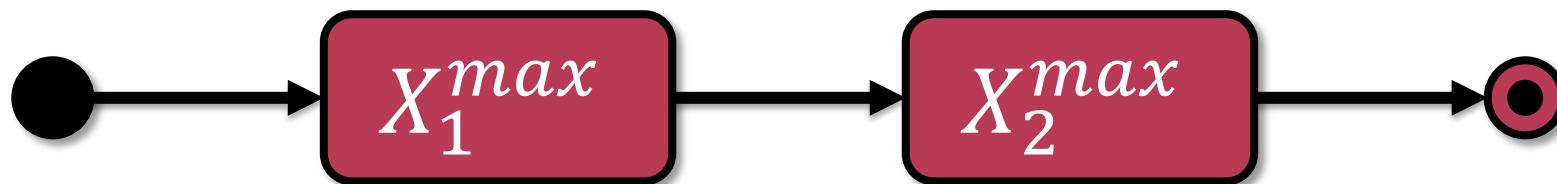
$$\chi^{max} = \min(\chi_1^{max}, \chi_2^{max})$$

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

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

Szekvenciális komponálás

χ^{max}

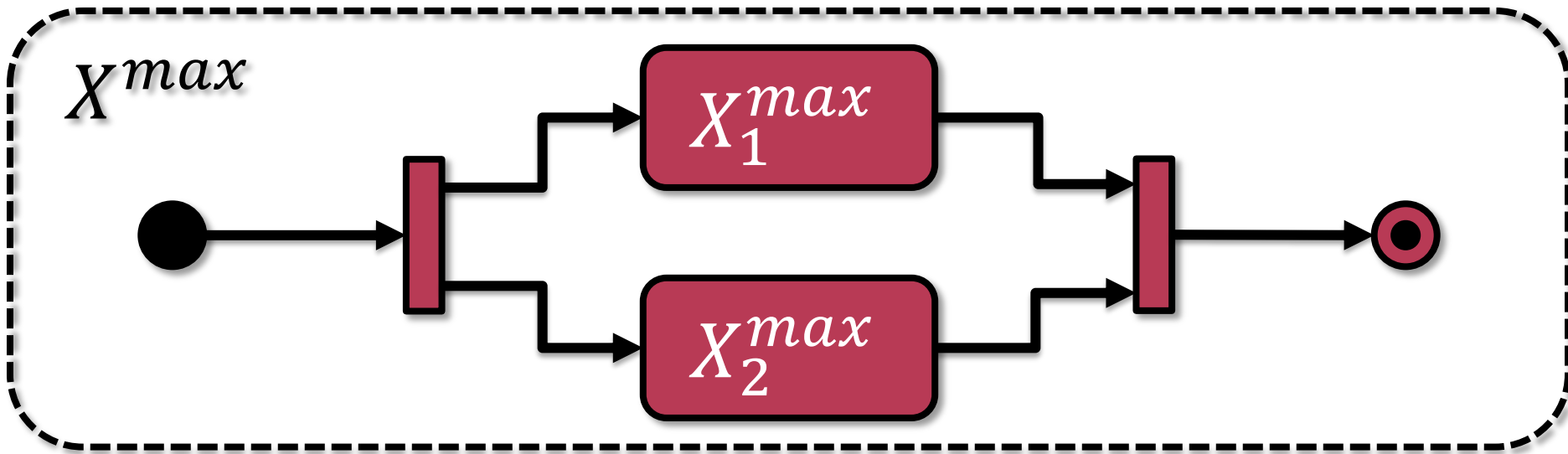


$$\chi^{max} = \min(X_1^{max}, X_2^{max})$$

Szűk keresztmetszet:

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

Párhuzamos komponálás

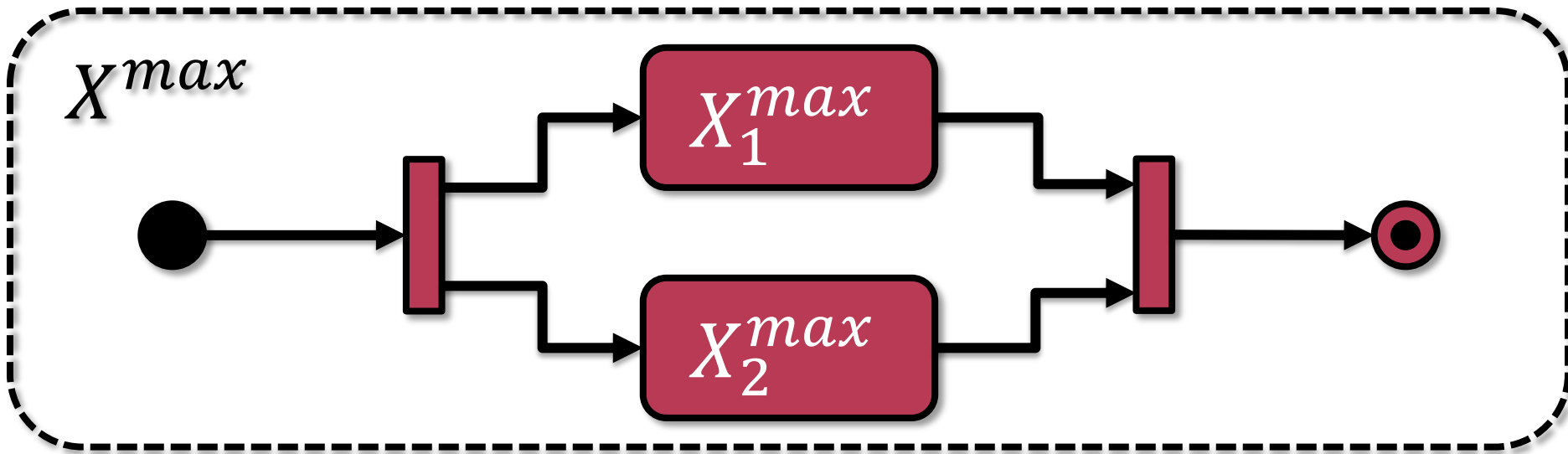


$$X^{max} = \min(X_1^{max}, X_2^{max})$$

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

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

Párhuzamos komponálás

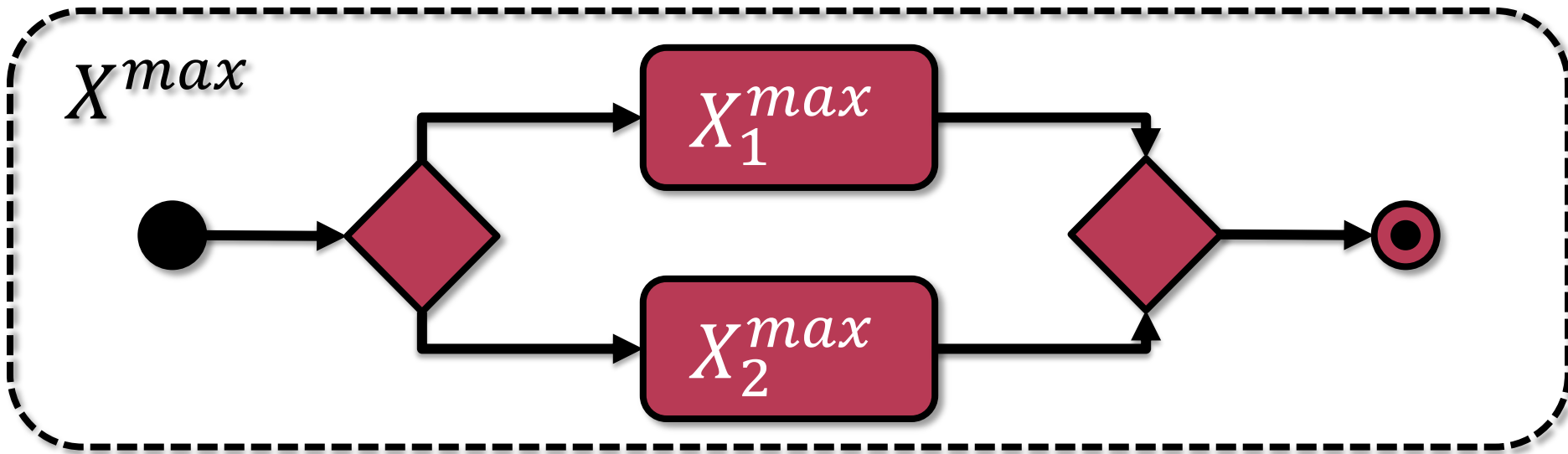


$$X^{max} = \min(X_1^{max}, X_2^{max})$$

Szűk keresztmetszet:

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

Komponálás szabad választással

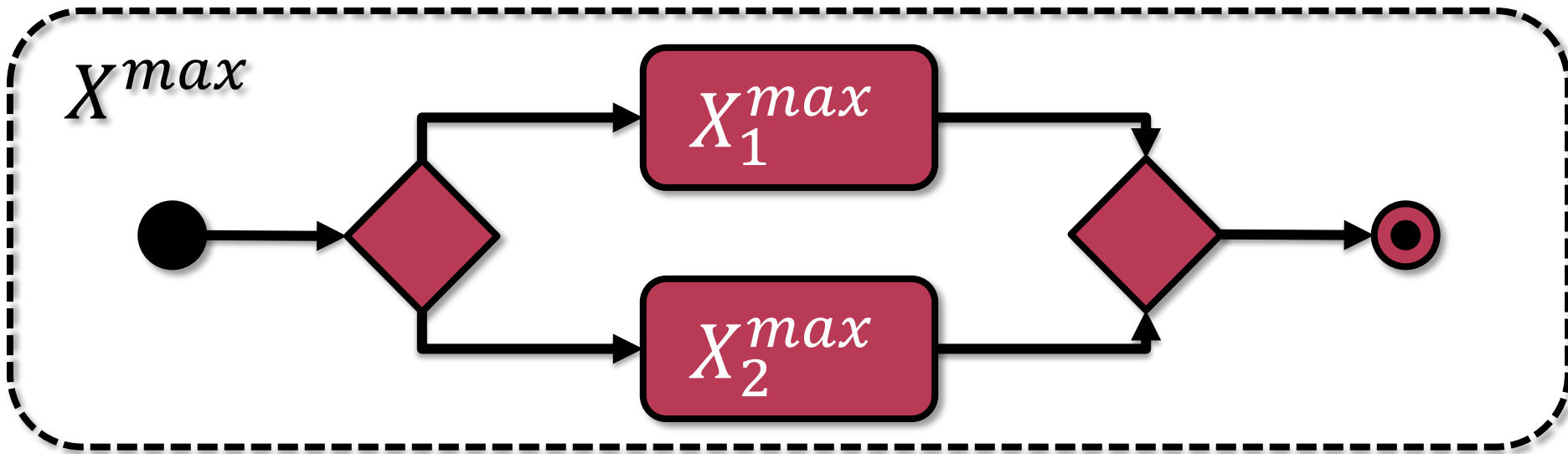


$$\chi^{max} = \chi_1^{max} + \chi_2^{max}$$

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

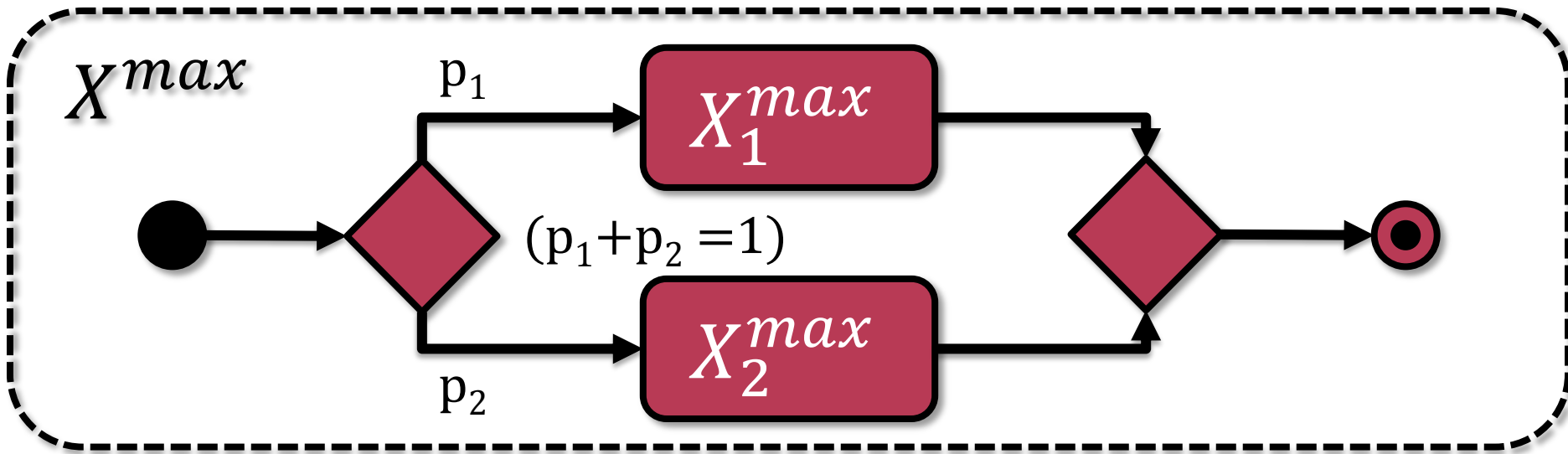


$$x^{max} = x_1^{max} + x_2^{max}$$

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

Feltétel: minden erőforráson ugyanannyi ideig tartson

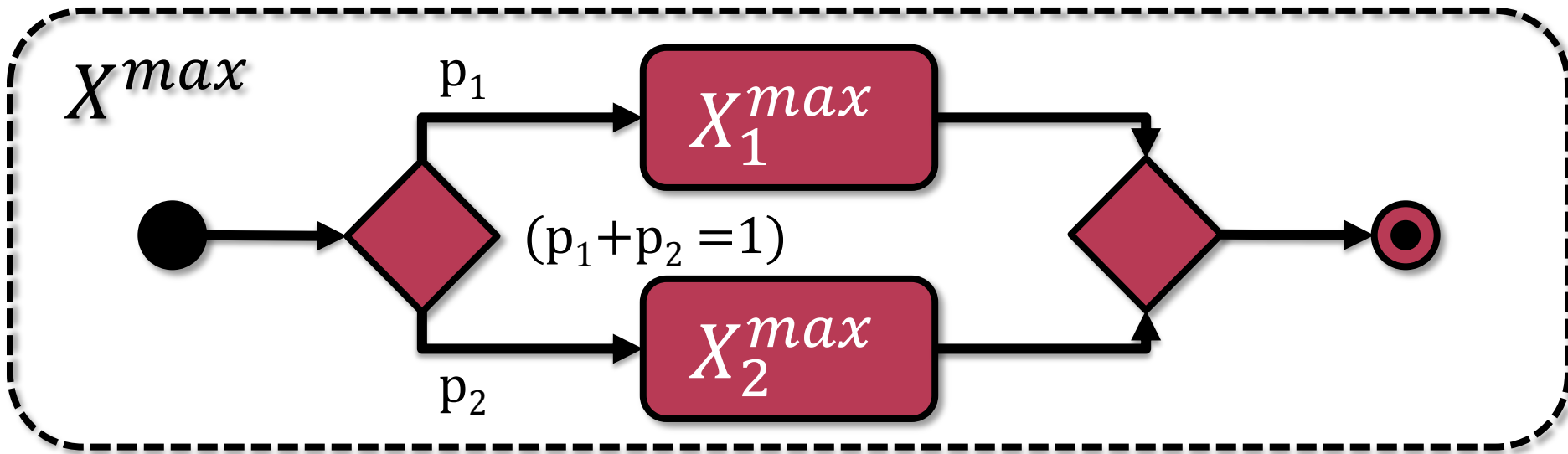
Komponálás kötött arányú választással



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

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.

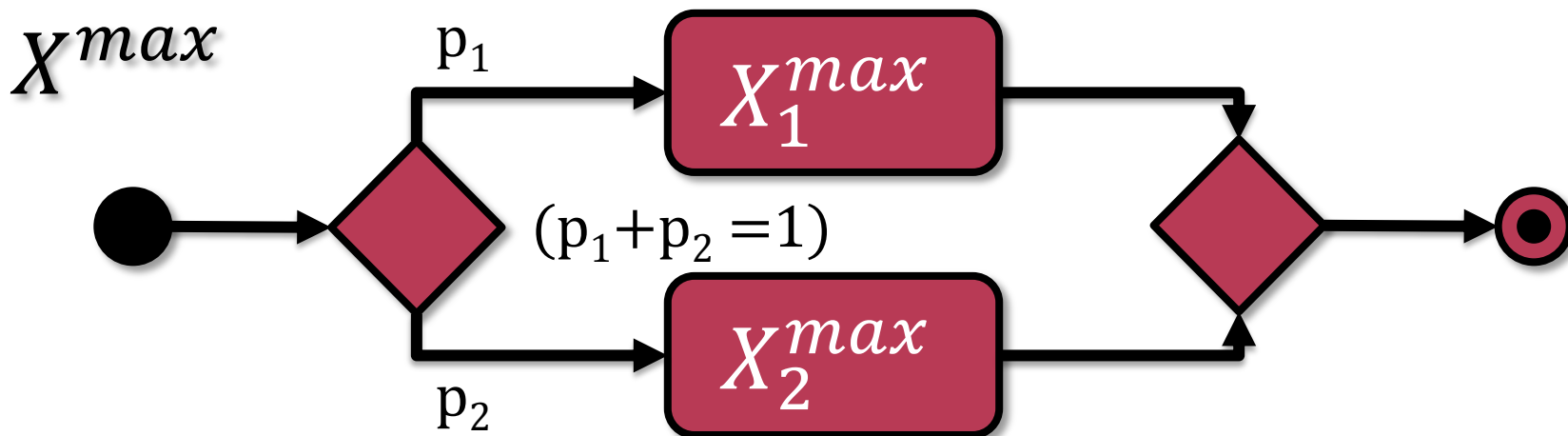
Komponálás kötött arányú választással



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

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

Komponálás kötött arányú választással



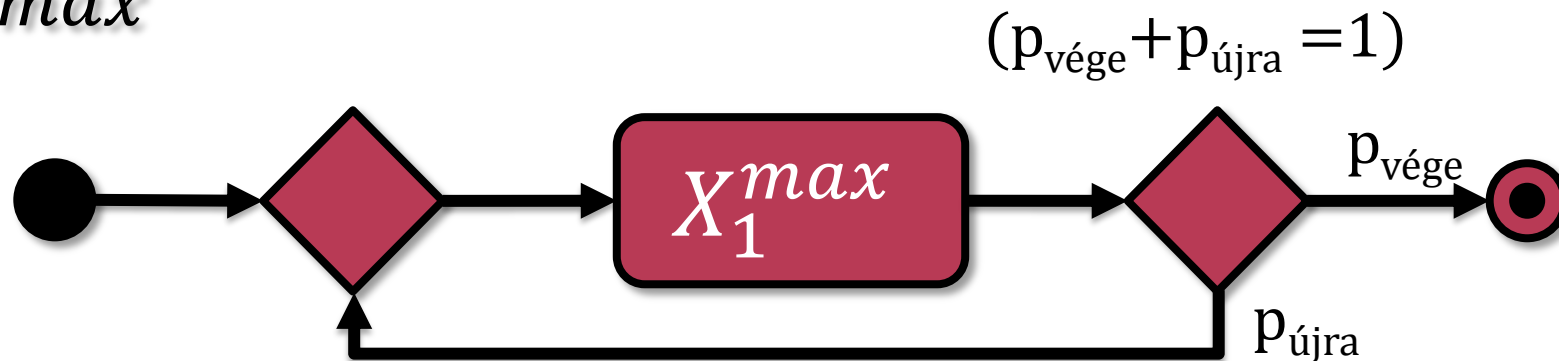
$$X^{max} = \min\left(\frac{1}{p_1} \times X_1^{max}, \frac{1}{p_2} \times X_2^{max}\right)$$

Szűk keresztmetszet:

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

Komponálás ciklussal

X^{max}

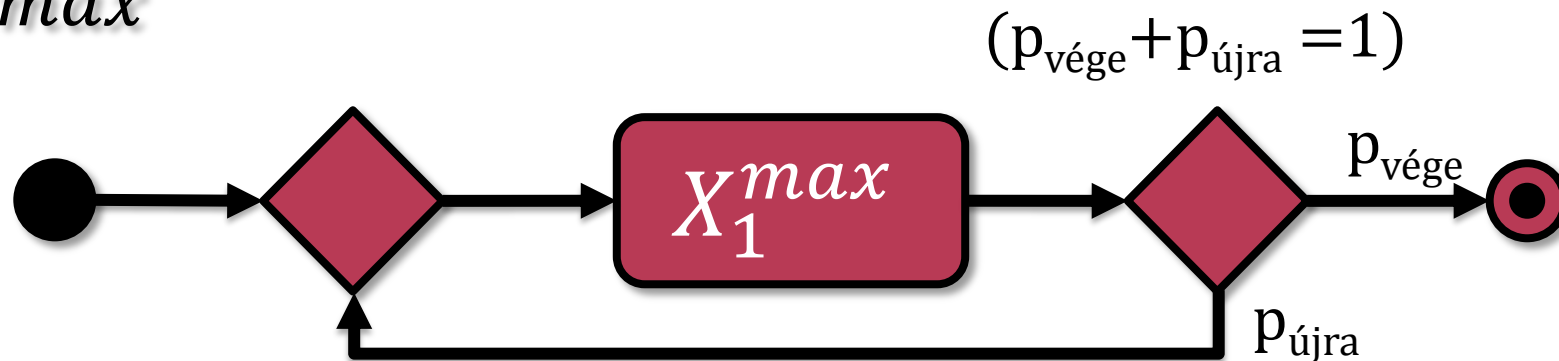


$$X^{max} = \frac{1}{p_{vége}} \times X_1^{max} = p_{vége} \times X_1^{max}$$

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:

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

- **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

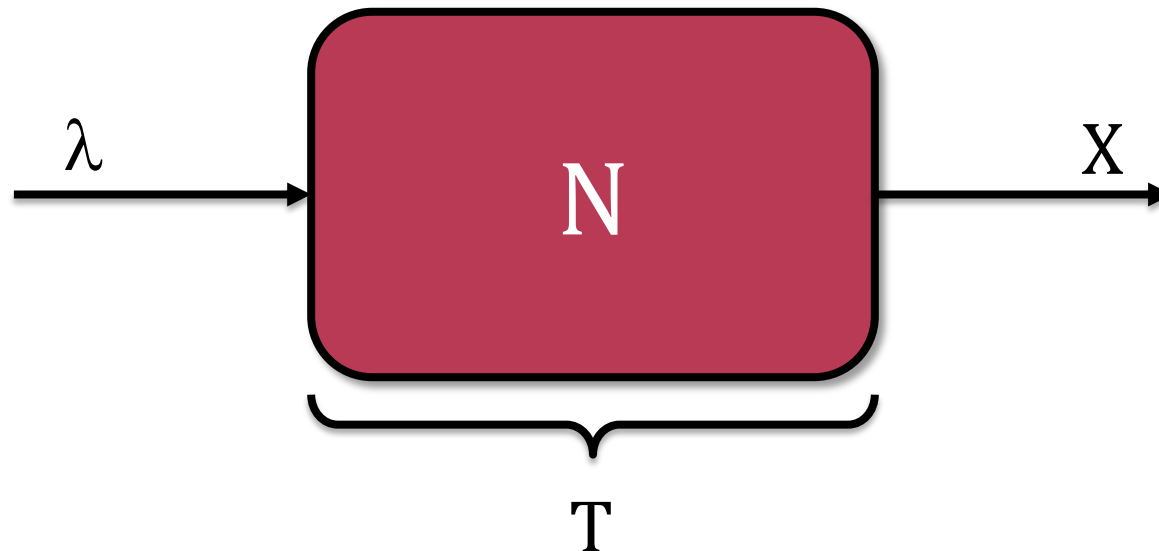
Terhelés változása

A LITTLE-TÖRVÉNY

Avagy az alapképlet

A Little-törvény

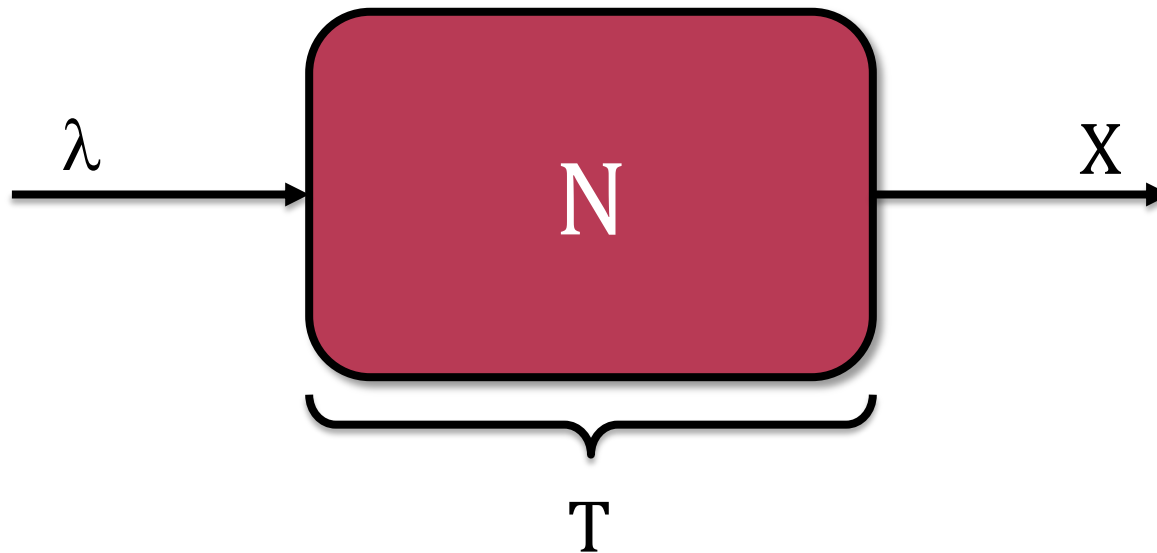
- λ : érkezési ráta $\left[\frac{darab}{s}\right]$
- X : átbocsátás $\left[\frac{darab}{s}\right]$
- T : rendszerben töltött idő $[s]$
- N : rendszerben lévő tokenek száma $[darab]$



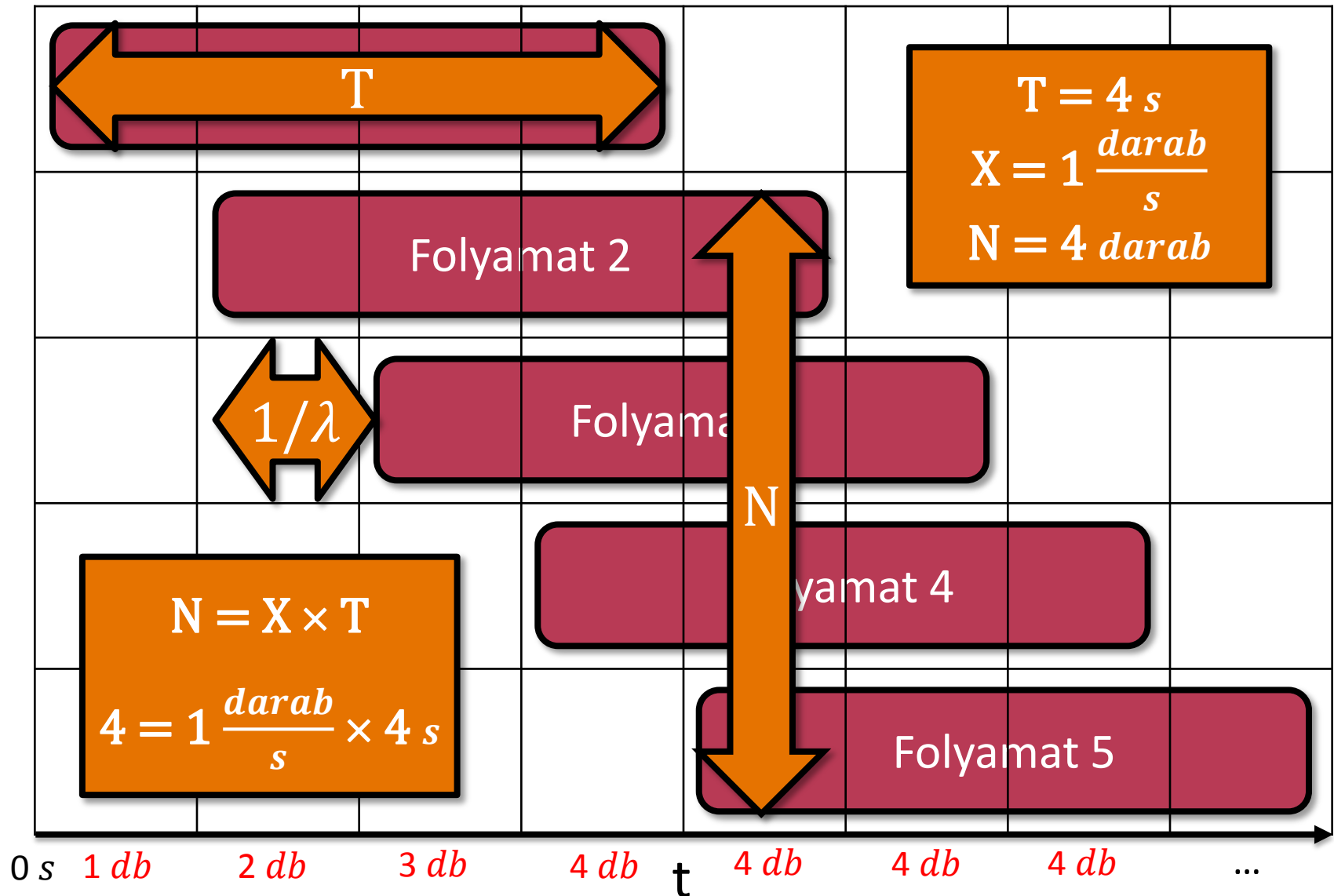
A Little-törvény

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

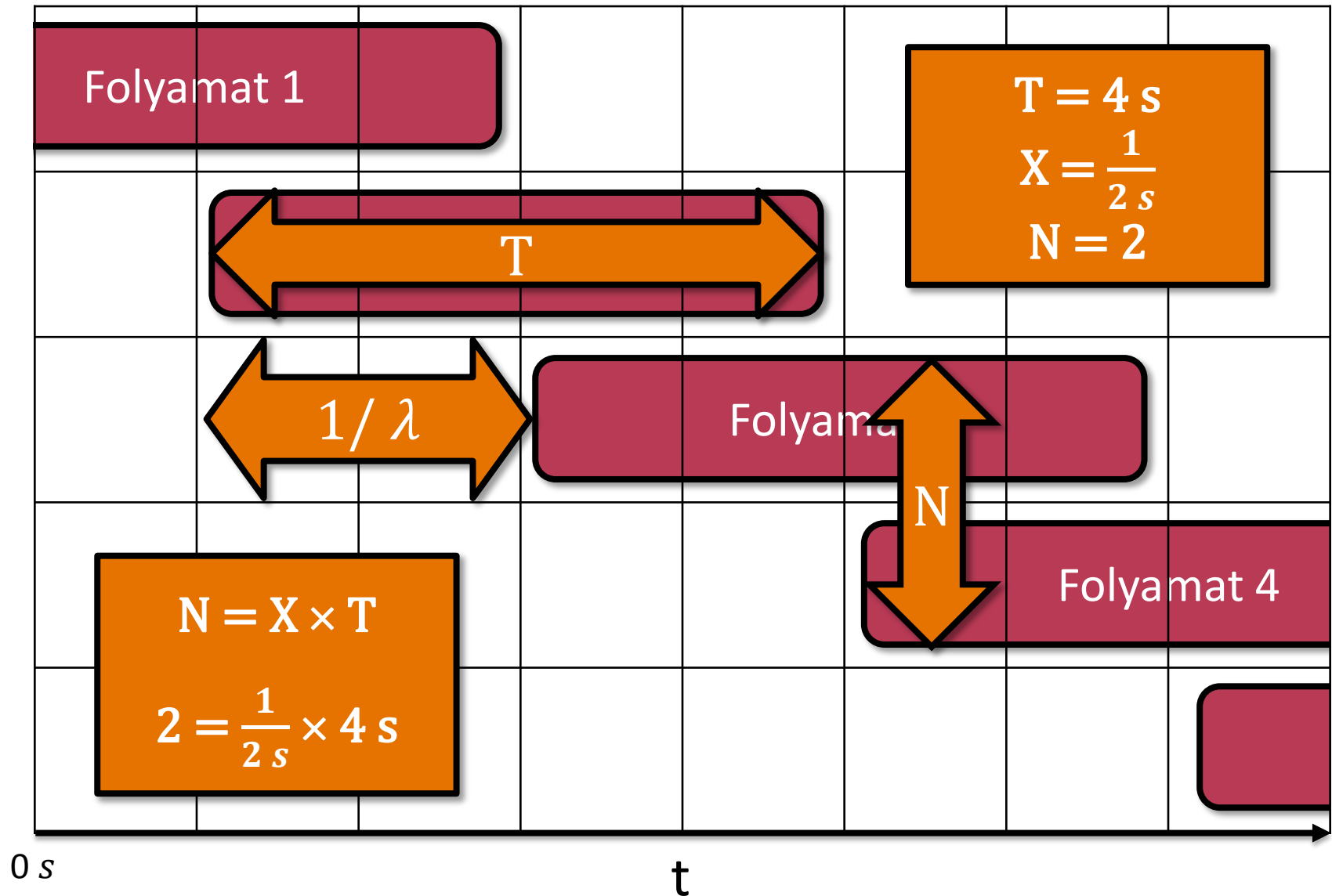
$$N = X \times T$$



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

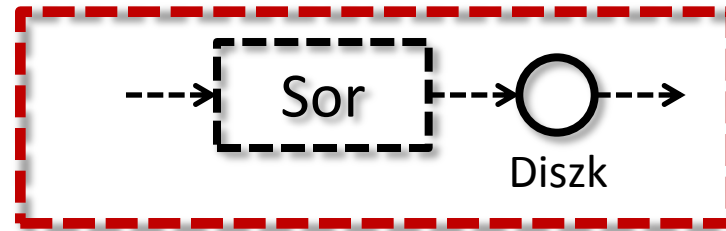


- Erőforrás: diszk
- 40 kérést szolgál ki másodpercenként (nincs átlapolódás)
- 1 kérés kiszolgálása átlagosan 0,0225 másodpercig tart
- Mekkora a kihasználtság?

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

Példa

Rendszer



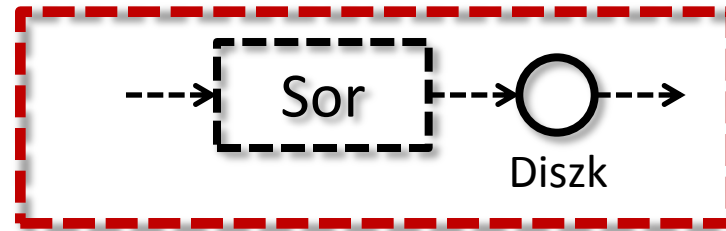
- Sorban állás is van a diszk előtt
- Diszk: 40 *kérés/s*
- Kérések átlagos száma a rendszerben: 4

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

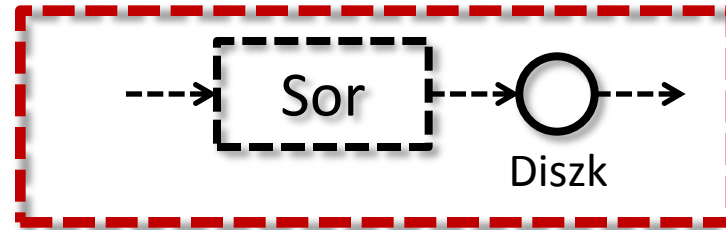
$$N = X \times T \rightarrow T_{rendszer} = 4 \text{ *kérés*} / 40 \frac{\text{kérés}}{s} = 0,1 \text{ s}$$

Átlagos sorban állási idő?

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

Példa

Rendszer



- 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>

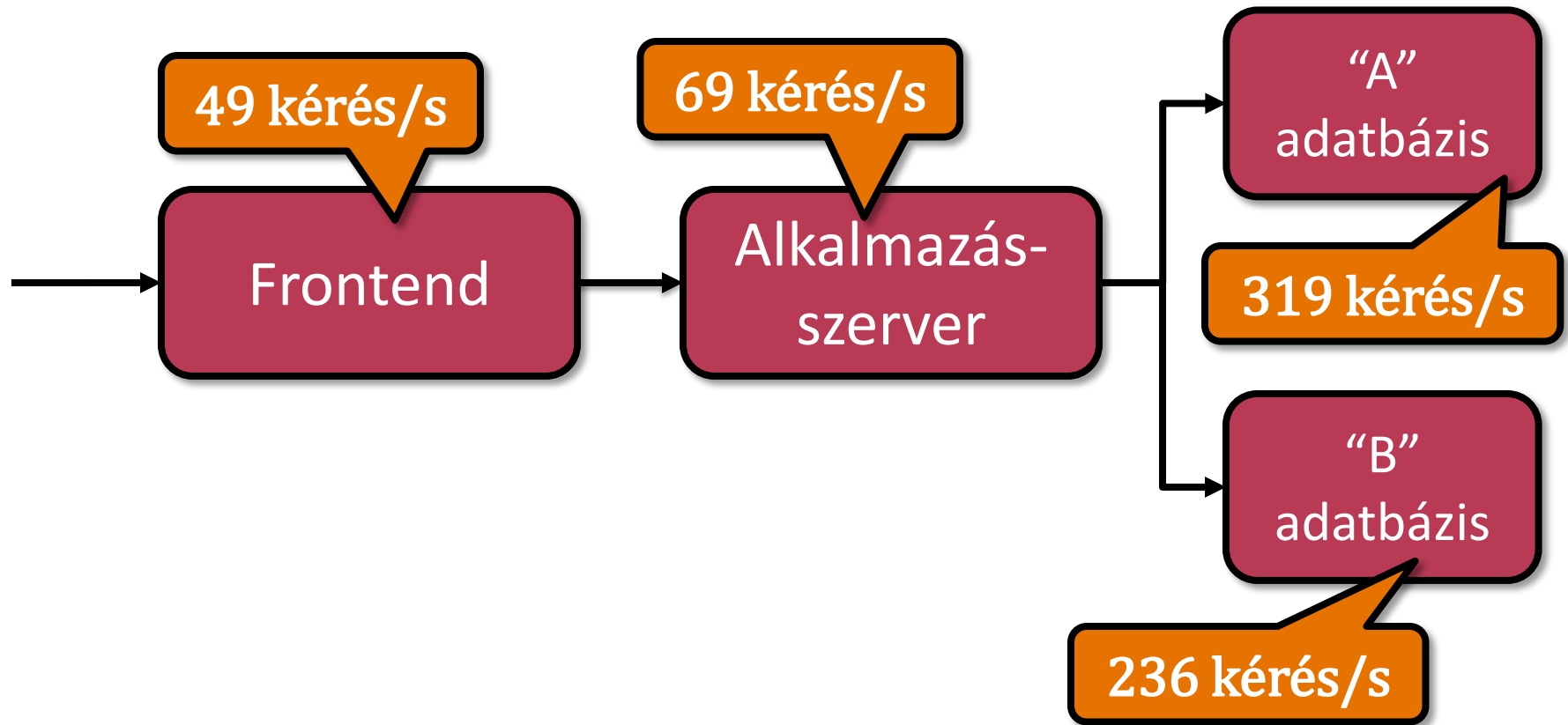
■ Miért oktatják(juk)

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

■ Példák

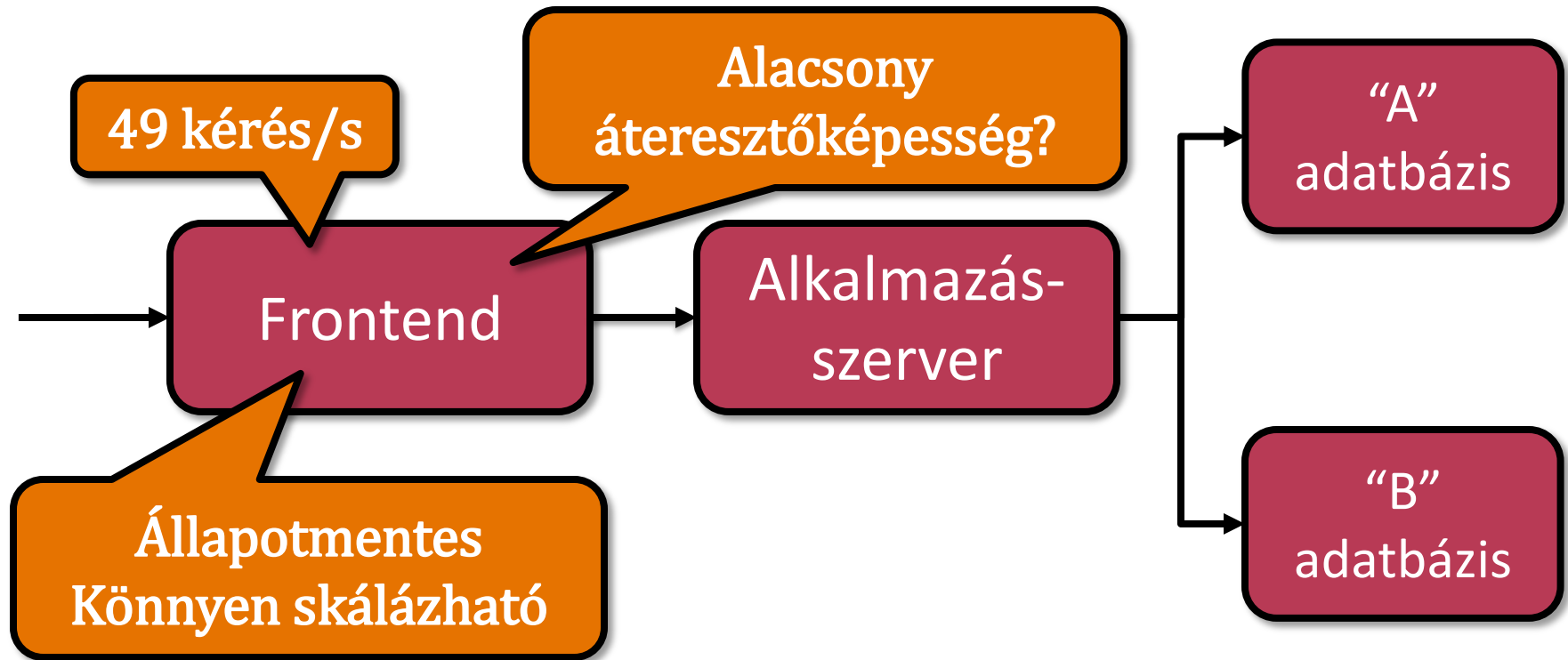
- <http://web.mit.edu/sgraves/www/papers/Little's%20Law-Published.pdf>

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



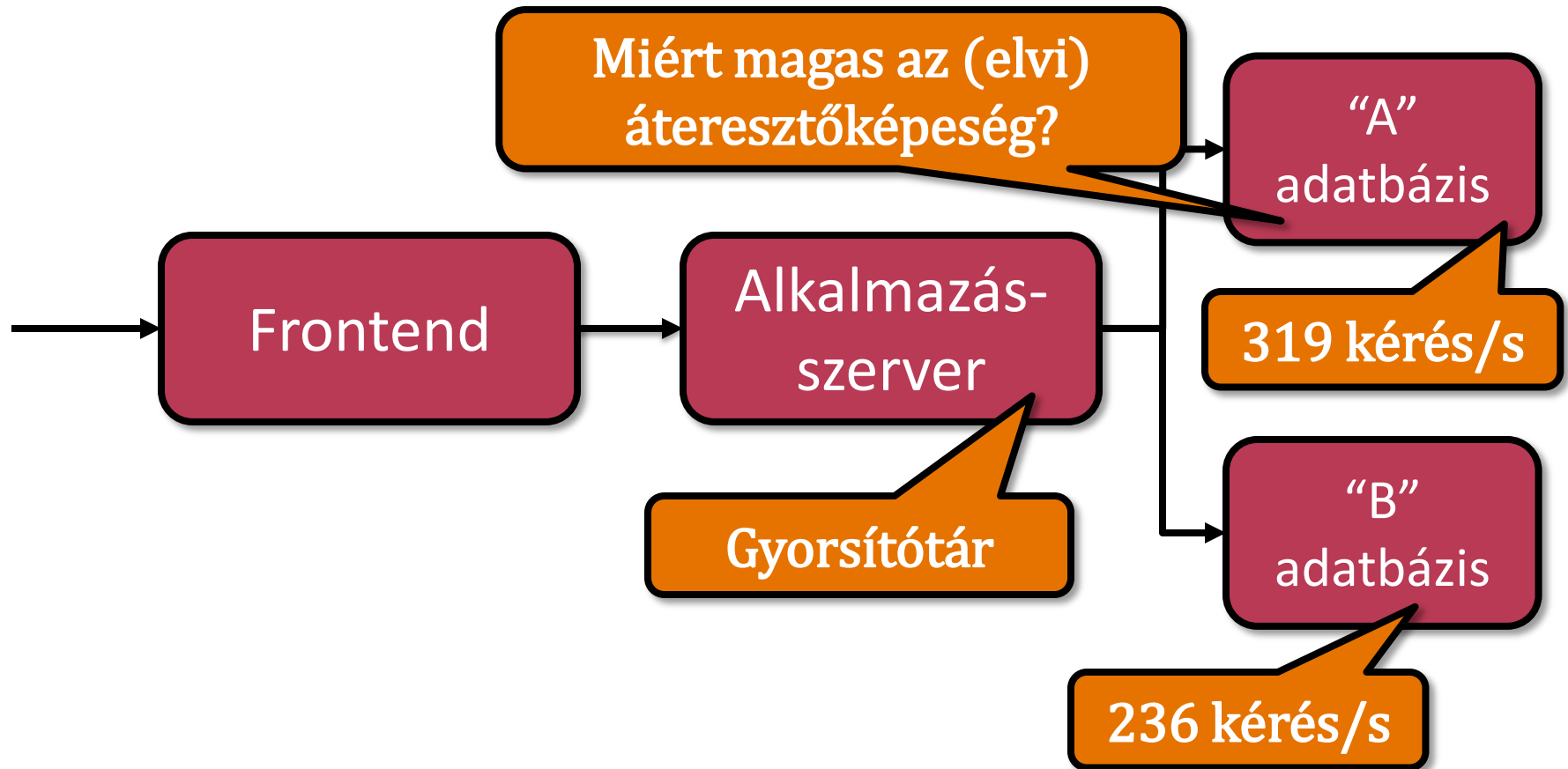
A mérőszámok itt az egész rendszerre érkező terhelésre utalnak! Például az „A. adatbázis” akkor válik szűk keresztmetszetté, ha a rendszerbe 319 lekérdezés érkezik másodpercenként.

Teljesítmény 3 rétegű architektúrában



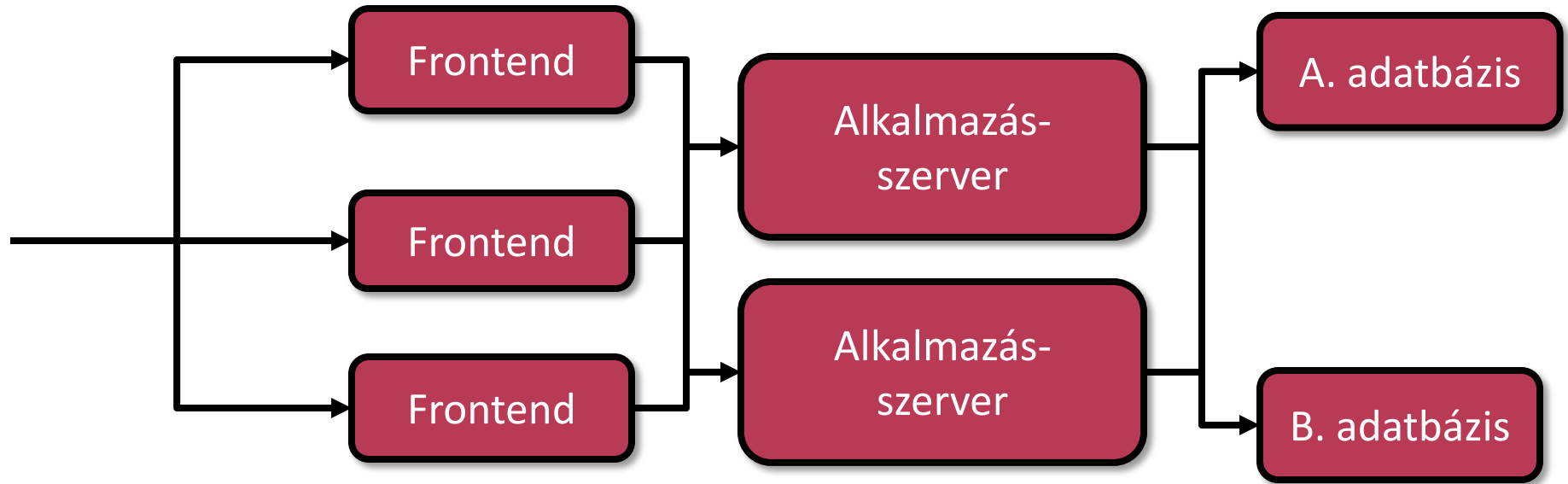
A mérőszámok itt az egész rendszerre érkező terhelésre utalnak! Például az „A. adatbázis” akkor válik szűk keresztmetszetté, ha a rendszerbe 319 lekérdezés érkezik másodpercenként.

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:)

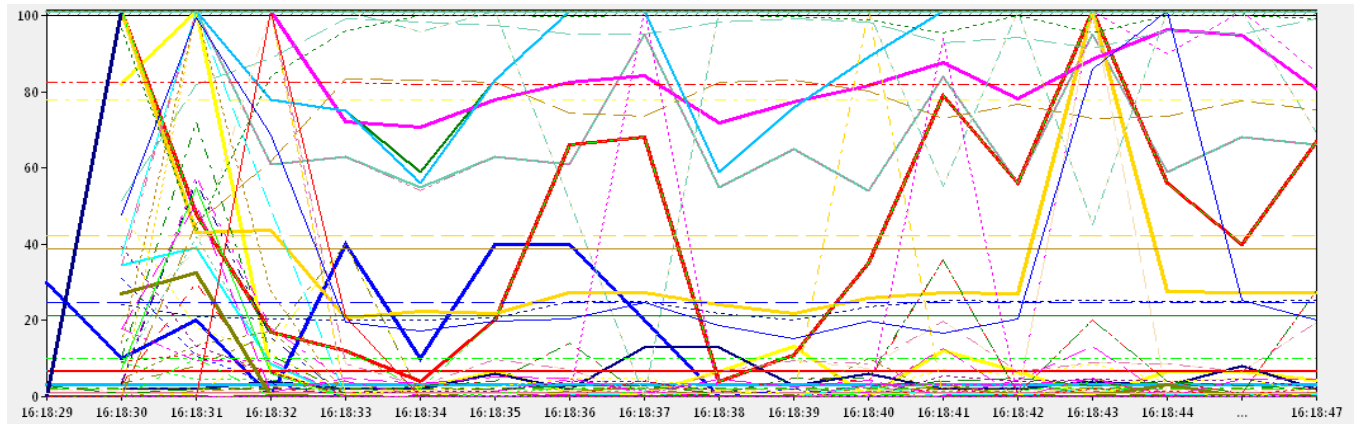
<http://www.projectclearwater.org/wp-content/uploads/2013/05/Clearwater-Deployment-Sizing-10-Apr-13.xlsx>

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

Mit mérjük/mi a lényeges?

- Metrikák “kicsiben”

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

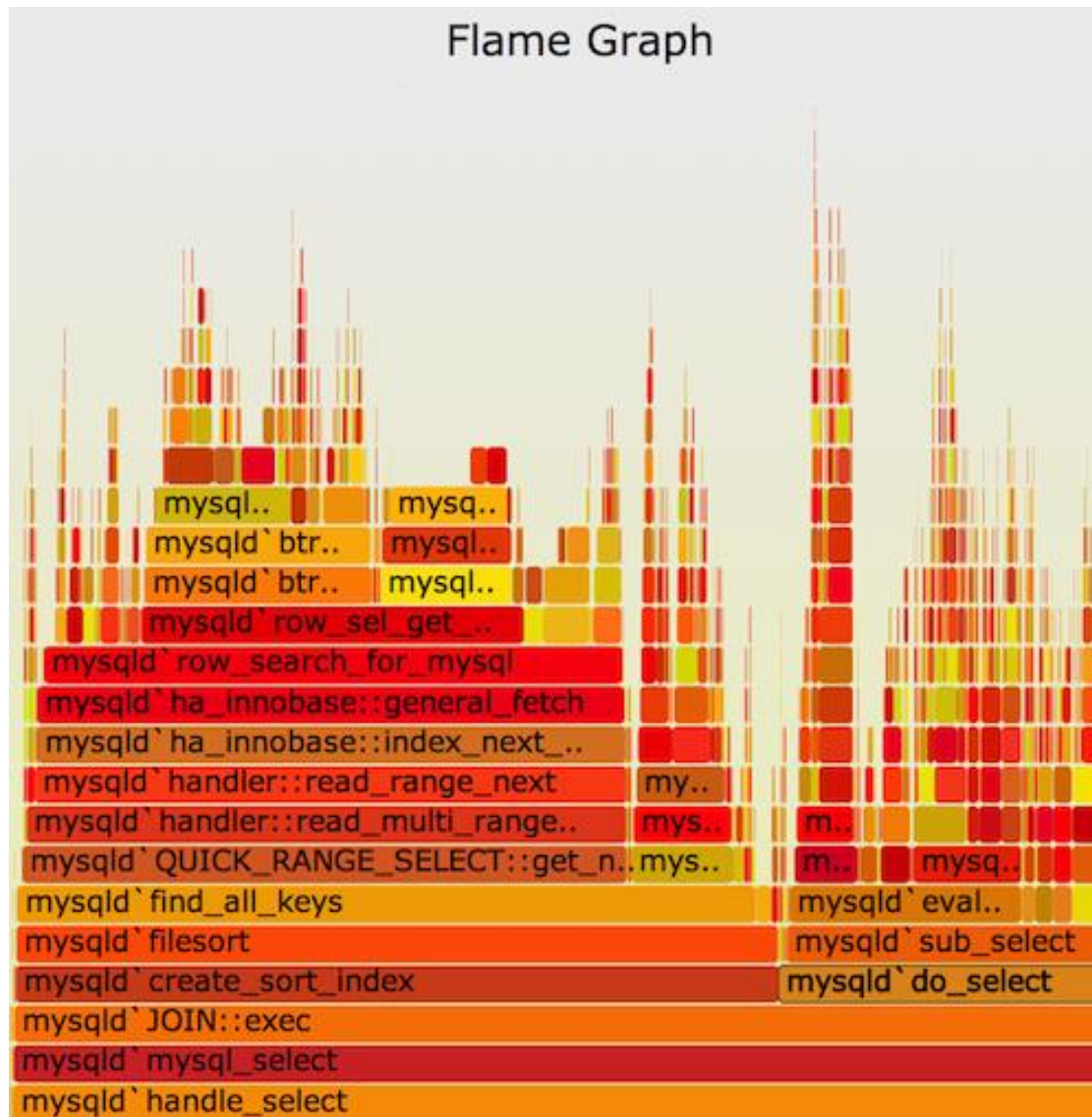


- Metrikák “nagyban”

- Pl. virtualizált rendszer

- Melyik az érdekes?

Példa: mit számol ez a gép enyit?



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

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



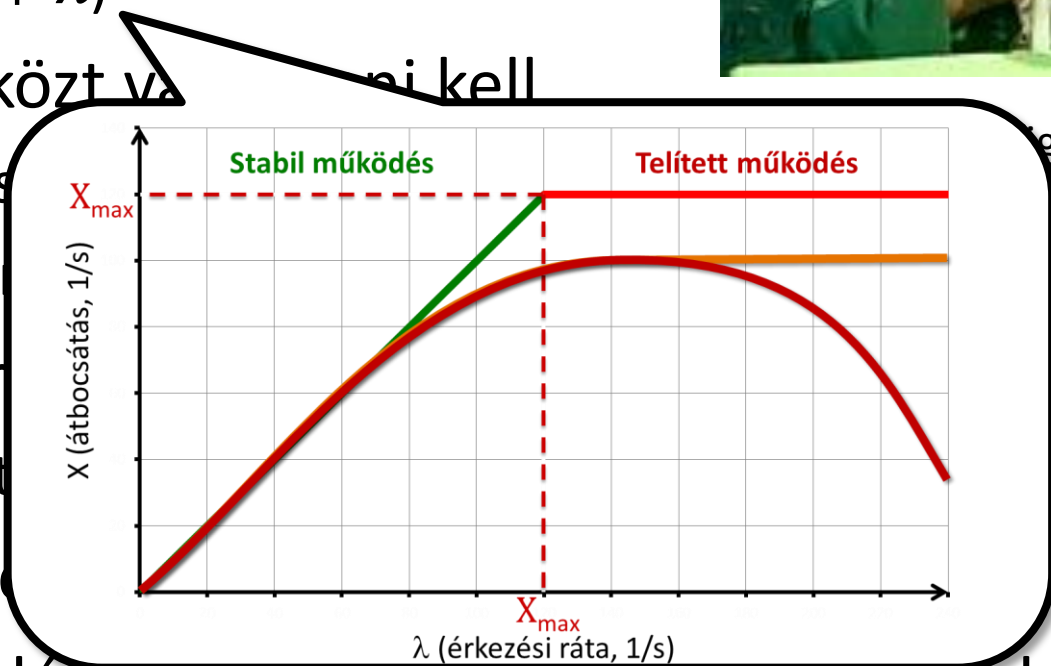
“A kezemet figyeljék, mert...”
(kép: wikipedia)

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



- terheléselos
 - Pl. ugyan
 - Konkrét beér
 - Pont ez a Lit
 - Egy feladat v
 - A rendszer felépítése/parameterai változhatnak
- figyeljük, mert...”
a)
ünk



A Little-törvény

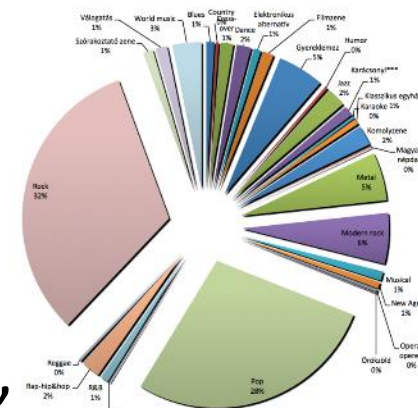
A Zipf-törvény

Terhelés változása

TERHELÉSMODELLEK: ZIPF TÖRVÉNY

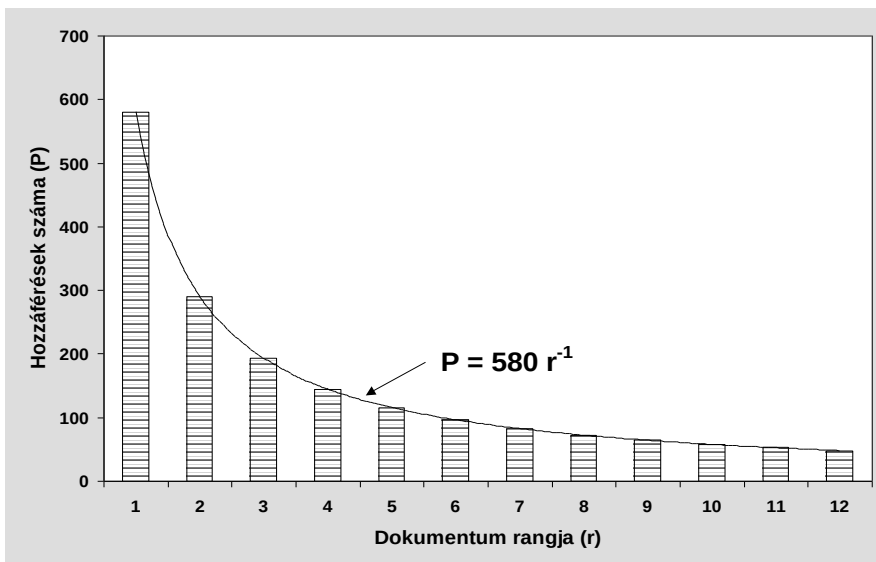
Mi a kérések tartalma?

- Eddig: minden kérés egyforma
 - “kérem egy könyv adatait”
- Valójában: a kéréseknek tartalmuk van
 - “Kérem az *Alapítány és Birodalom* adatait”
 - Ld. Pareto elv (80% -- 20%)
 - A kérések (többsége) az adatok (kis részére) irányul
- Lényeges, mert
 - Műszaki hatása van
 - Cache, pool size, statikus tár, ...
 - A rendszermodellt is érinti
 - Gyakori kéréseket másképp kezeljük



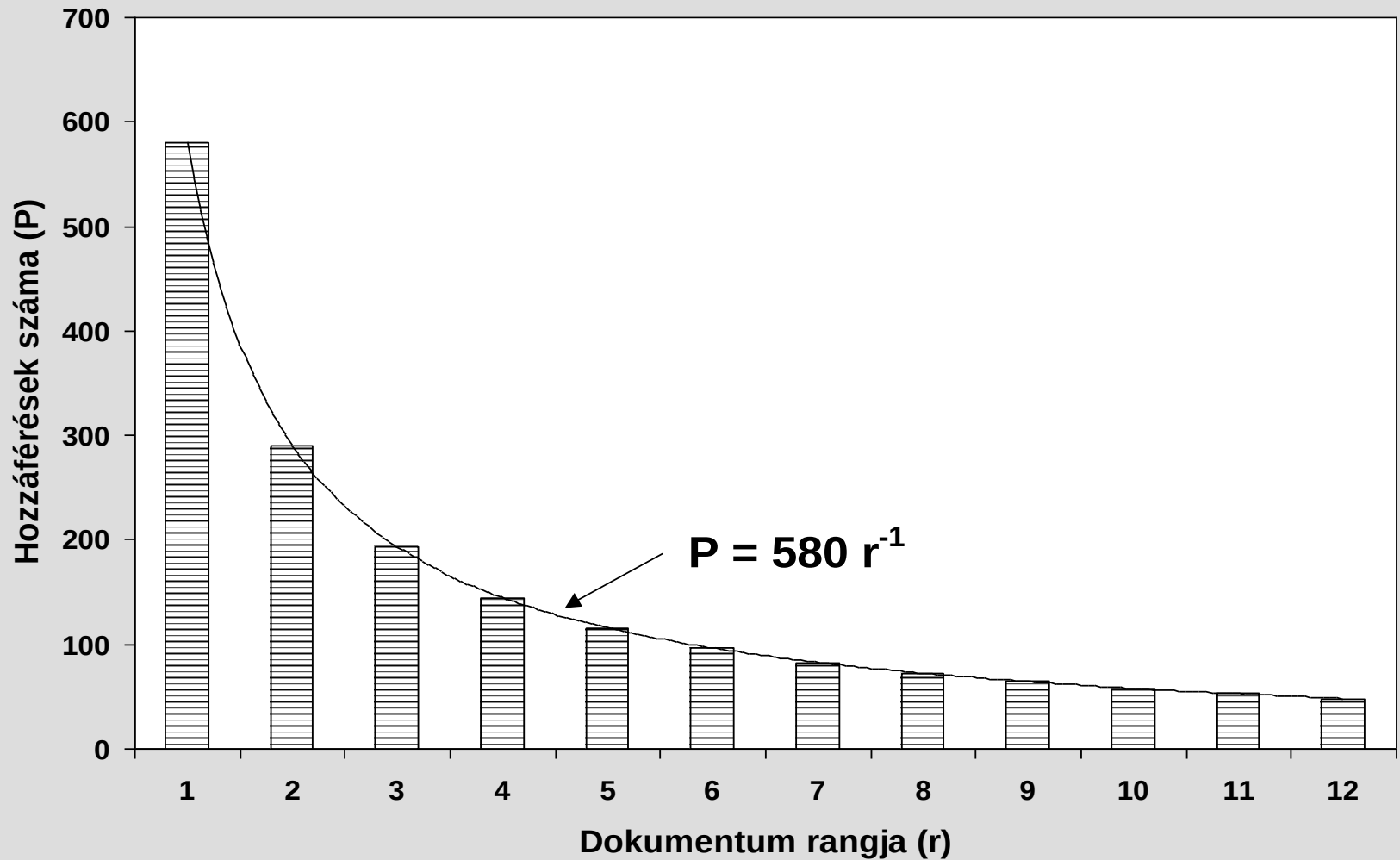
Zipf törvénye

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



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

Zipf – Példa (1)



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)

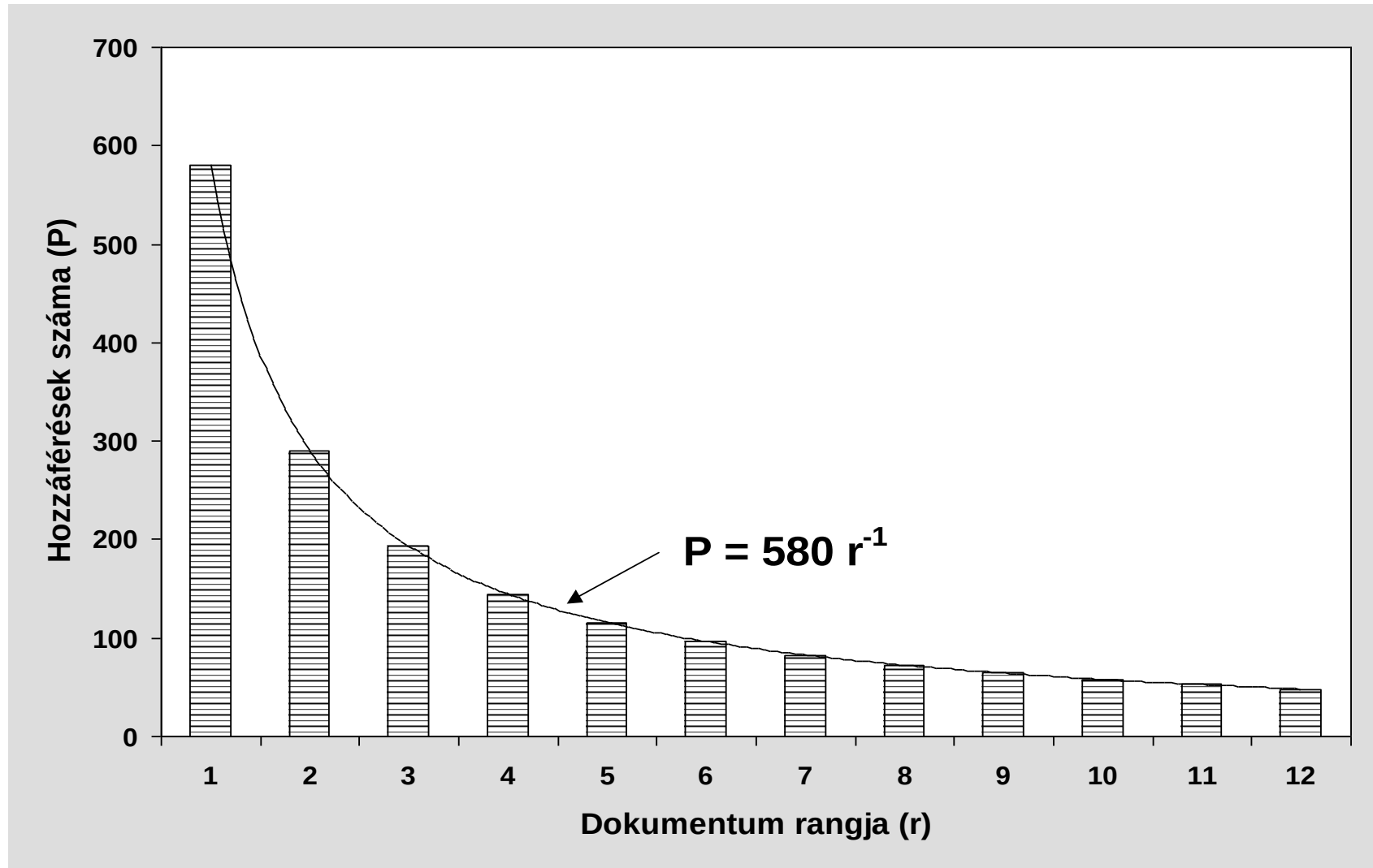
Zipf törvénye – pl. Web dokumentumokra

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

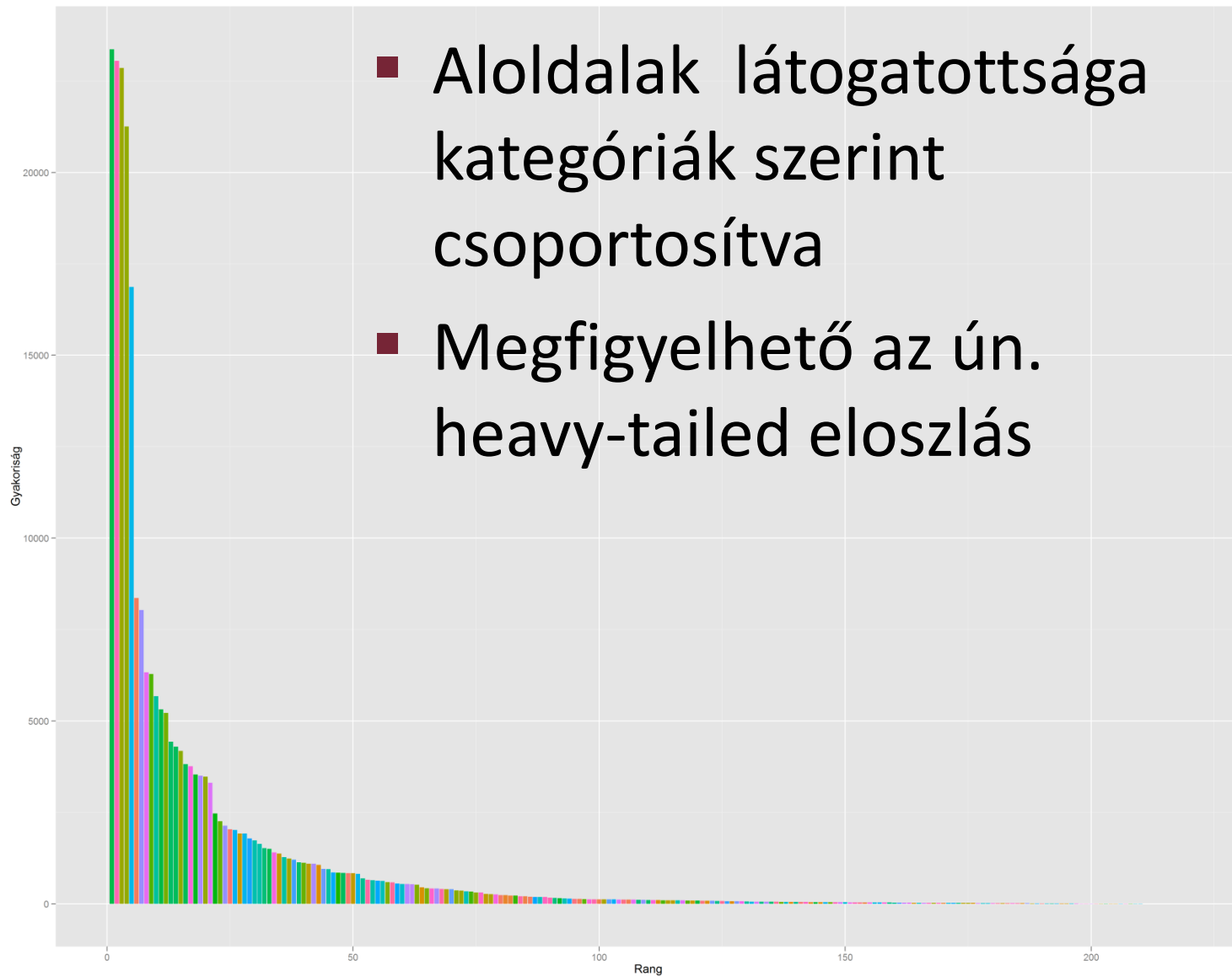
- P – hivatkozások (elérések)
- r – rang (1 = leggyakoribb)
- k – pozitív konstans

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

Zipf – Példa (1)



Zipf – Példa: Tanszéki honlap (1)

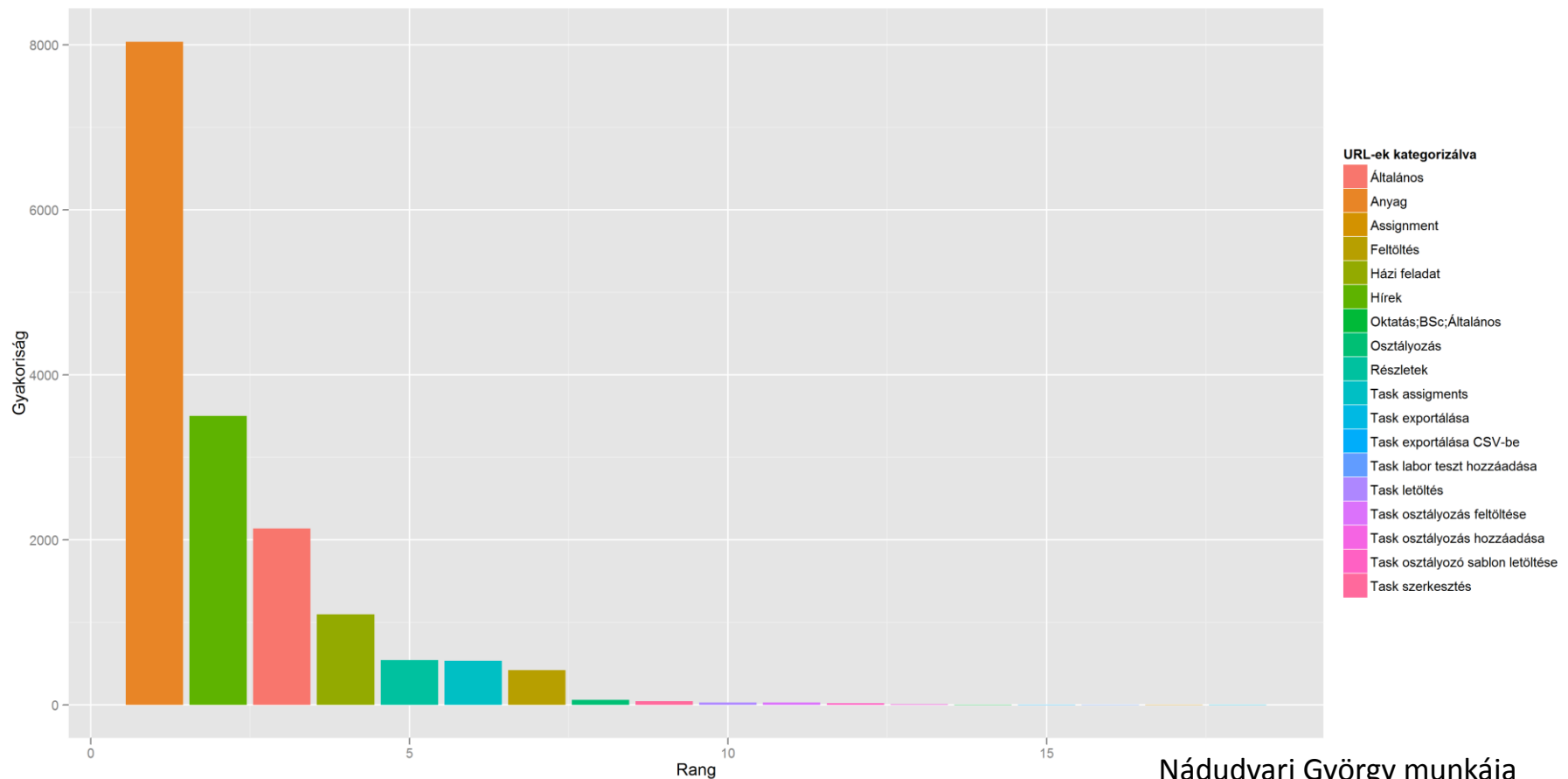


URL-ek kategorizálva

1	...
2	...
3	...
4	...
5	...
6	...
7	...
8	...
9	...
10	...
11	...
12	...
13	...
14	...
15	...
16	...
17	...
18	...
19	...
20	...
21	...
22	...
23	...
24	...
25	...
26	...
27	...
28	...
29	...
30	...
31	...
32	...
33	...
34	...
35	...
36	...
37	...
38	...
39	...
40	...
41	...
42	...
43	...
44	...
45	...
46	...
47	...
48	...
49	...
50	...
51	...
52	...
53	...
54	...
55	...
56	...
57	...
58	...
59	...
60	...
61	...
62	...
63	...
64	...
65	...
66	...
67	...
68	...
69	...
70	...
71	...
72	...
73	...
74	...
75	...
76	...
77	...
78	...
79	...
80	...
81	...
82	...
83	...
84	...
85	...
86	...
87	...
88	...
89	...
90	...
91	...
92	...
93	...
94	...
95	...
96	...
97	...
98	...
99	...
100	...
101	...
102	...
103	...
104	...
105	...
106	...
107	...
108	...
109	...
110	...
111	...
112	...
113	...
114	...
115	...
116	...
117	...
118	...
119	...
120	...
121	...
122	...
123	...
124	...
125	...
126	...
127	...
128	...
129	...
130	...
131	...
132	...
133	...
134	...
135	...
136	...
137	...
138	...
139	...
140	...
141	...
142	...
143	...
144	...
145	...
146	...
147	...
148	...
149	...
150	...
151	...
152	...
153	...
154	...
155	...
156	...
157	...
158	...
159	...
160	...
161	...
162	...
163	...
164	...
165	...
166	...
167	...
168	...
169	...
170	...
171	...
172	...
173	...
174	...
175	...
176	...
177	...
178	...
179	...
180	...
181	...
182	...
183	...
184	...
185	...
186	...
187	...
188	...
189	...
190	...
191	...
192	...
193	...
194	...
195	...
196	...
197	...
198	...
199	...
200	...

Zipf – Példa: Tanszéki honlap (2)

- A Rendszermodellezés tárgy oldalainak látogatottsága



Nádudvari György munkája

A Little-törvény

A Zipf-törvény

Terhelés változása

TERHELÉS VÁLTOZÁSA

Milyen jellegű a terhelés?

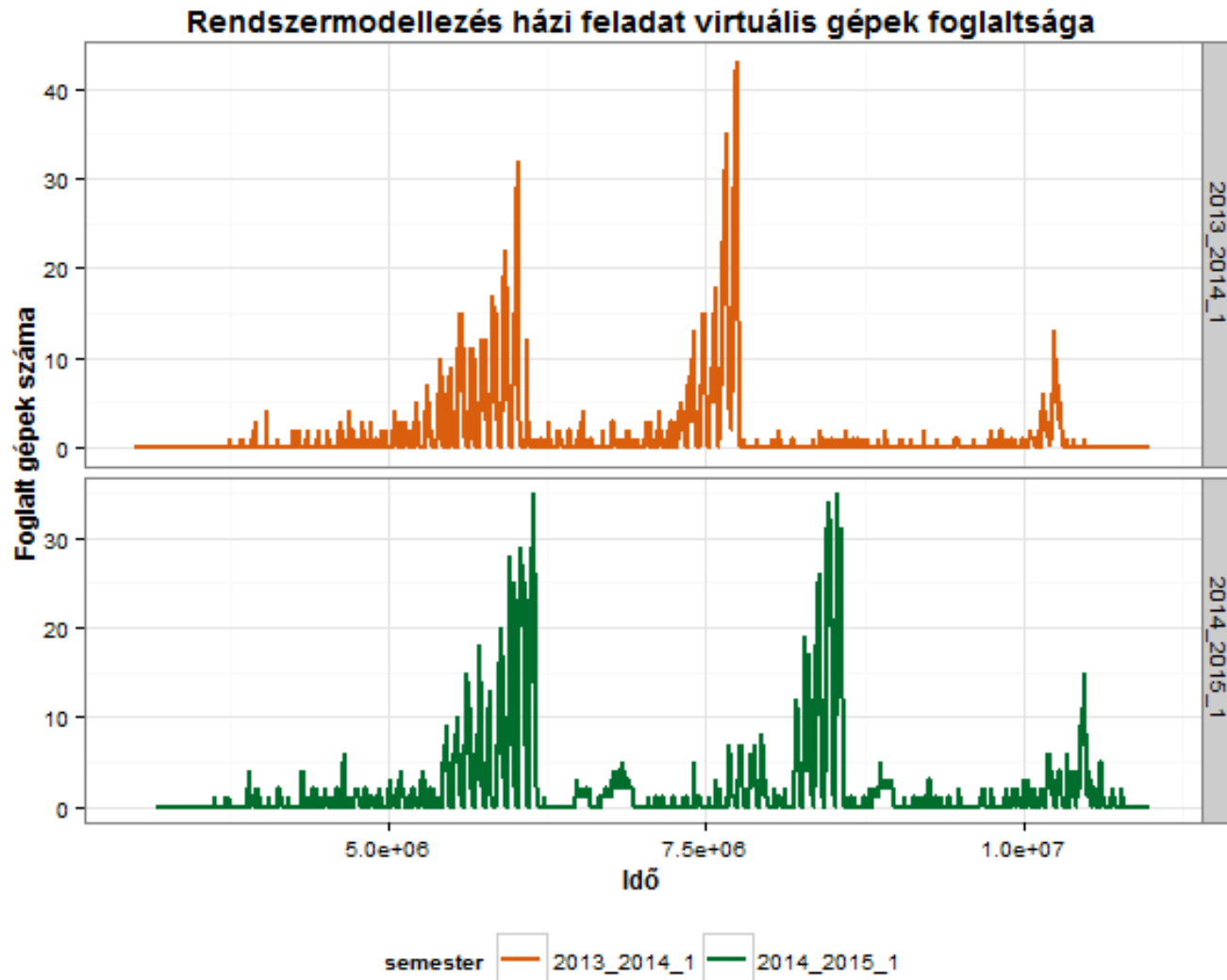
■ Eddig:

- Átlagos értékekkel számoltunk
- A rendszer viselkedését a *terhelés (intenzitás)* függvényében néztük
- De: valójában nem (feltétlenül) előre kiszámíthatóan változik a terhelés

■ Valójában

- A rendszer viselkedése *időben* változik
- Ennek műszaki hatásai vannak
 - Váltás feladatok közt, erőforrásfoglalás, stb. (pl. Operációs rendszerek)

Rendszermodellezés (7. félév) a felhőben



Rendszermodellezés (7. félév) a felhőben

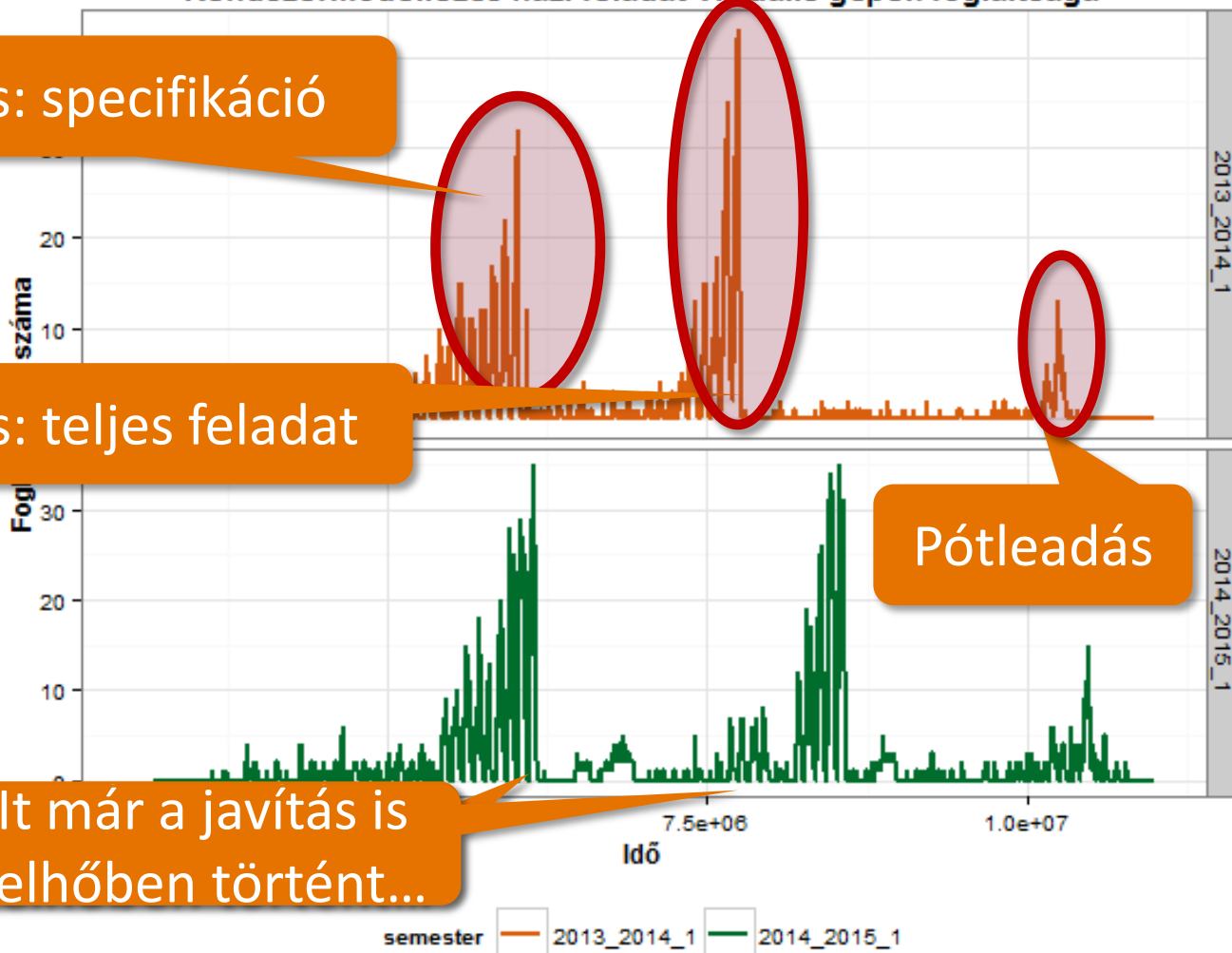
Rendszermodellezés házi feladat virtuális gépek foglaltsága

1. fázis: specifikáció

2. fázis: teljes feladat

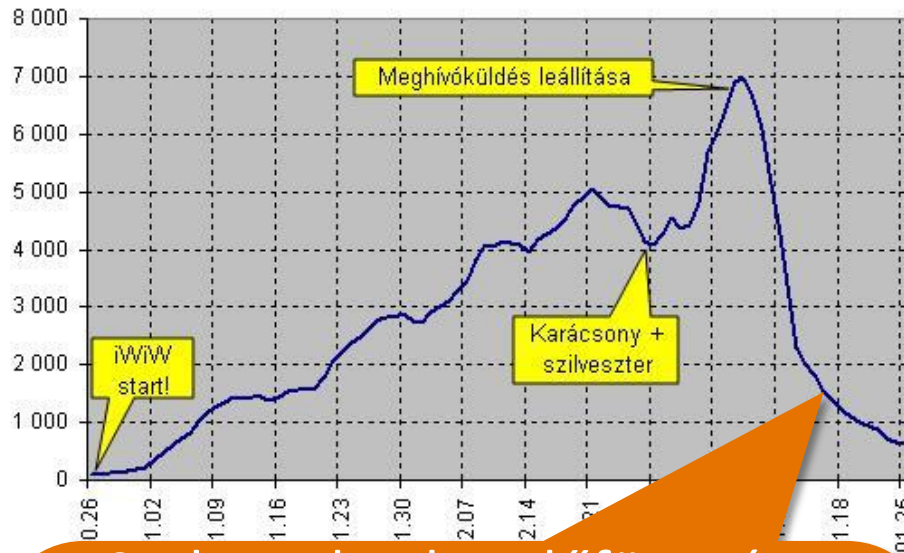
Pótleadás

It már a javítás is
felhőben történt...



Valós (történelmi) terhelés példa (iwiw)

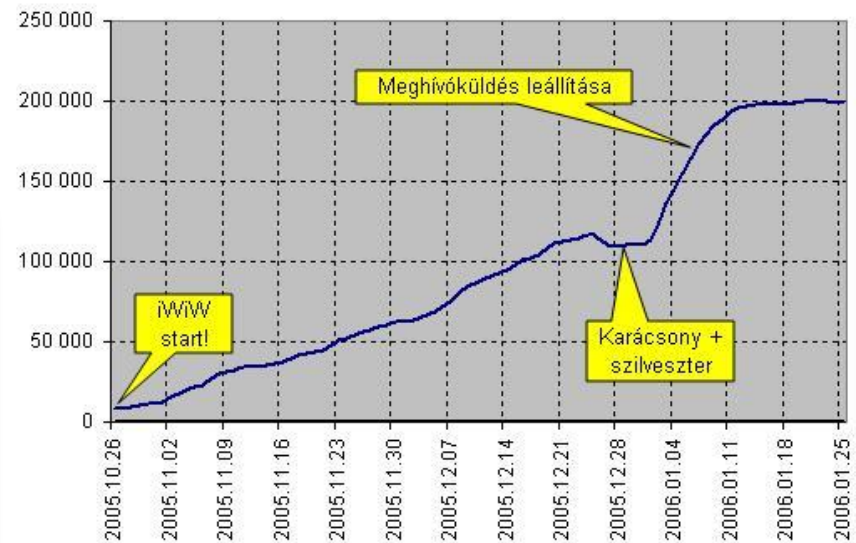
Napi regisztrációk (előző hét nap átlaga)



Szakaszokra becslőfüggvény
illeszthető

- Lineáris, exponenciális, logaritmikus
- Regresszió, Bővebben pl. Valószínűségyszámítás

Napi egyedi látogatók (előző hét nap átlaga)



Forrás: <http://www.sg.hu/cikkek/42924/>