# Leistungsmodellierung 2

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

# Wiederholung

- **Stabiler Zustand:**
  - kann mit Durchschnittswerten gerechnet werden
  - $\lambda = \mathrm{X}$ (Ankuftsrate = Durchsatz)
- **Grenzdurchsatz:**
  - der grösste erreichbare Durchsatz (bei T Durchschnittsabfertigungszeit)
  - $\mathrm{X}^{\max} = \frac{K}{T}$ (bei K frei wählbaren Ressourceninstanzen)
- **Auslastung**:
  - Verhältnis des aktuellen und des Grenzdurchsatzes
  - $\mathrm{U} = \frac{X}{K} \times \mathrm{T}$ (bei K frei wählbaren Ressourceninstanzen)

MŰEGYETEM 1782

Besichtigunszahl

Der Satz von Little

Der Satz von Zipf

Änderungen der Last

# INHALT

Besichtigunszahl | Der Satz von Little | Der Satz von Zipf | Änderungen der Last

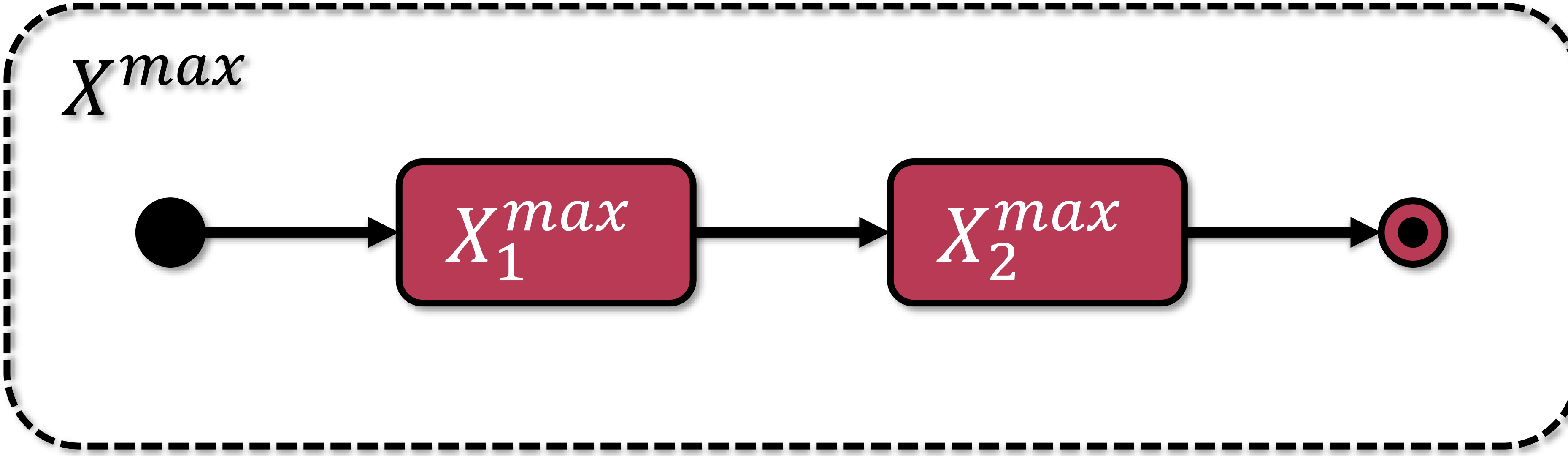
# GRENZDURCHSATZ UND BESICHTIGUNGSZAHL

# Durchsatz der Prozessmodelle

- Den Aktivitäten werden Ressourcen zugeordnet
  - Die (durchschnittl.) Ausführungszeit ist auch gegeben
    → $X^{max}$ der Aktivitäten kann berechnet werden
- Z.B. im Neptun System belastet die Anmeldung zu einer LVA den DB-Server 100 ms lang
  - $\mathrm{T} = 100\ \mathrm{ms}$
  - $X^{max} = \frac{1}{T} = 10\ \frac{Anmeldung}{Sec}$

**Wie hoch ist der Grenzdurchsatz eines Systems, wenn der Prozess (z.B. das Verhalten des Benutzers) gegeben ist?**

# Sequentielle Komposition

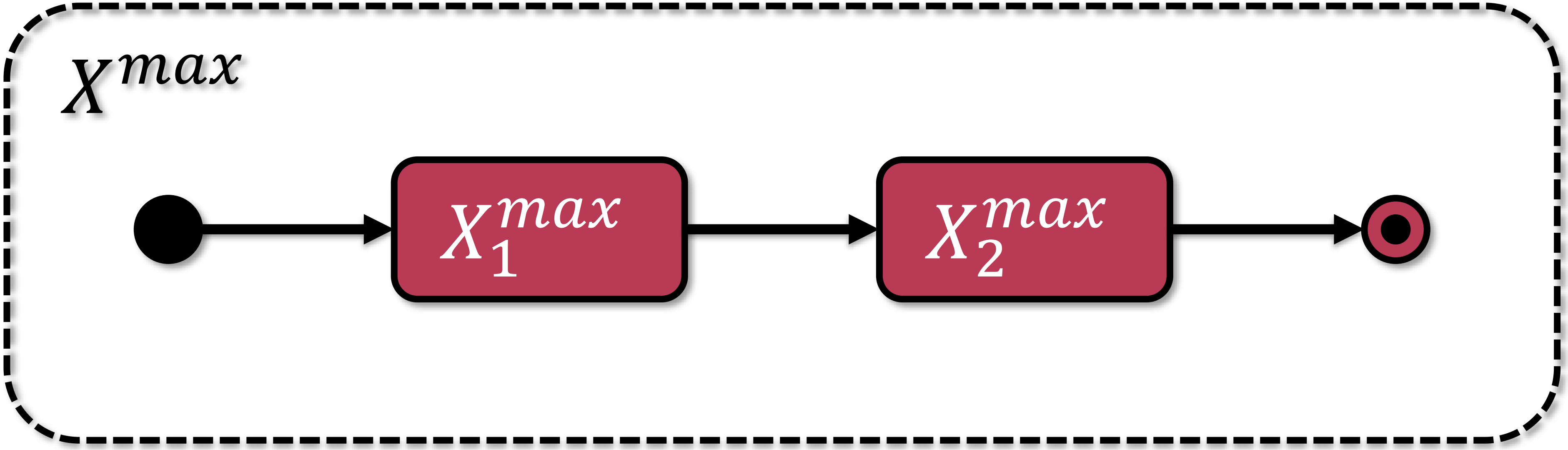



Jede Aktivität wird einmal besichtigt.

Auch wenn die eine Aktivität schnell ausgeführt wird, häufen sich die Tokens vor der anderen an

z.B. im Bürgerbüro:
Nummer Ziehen (300/Stunde),
Fallbearbeitung (2/Stunde)

# Sequentielle Komposition

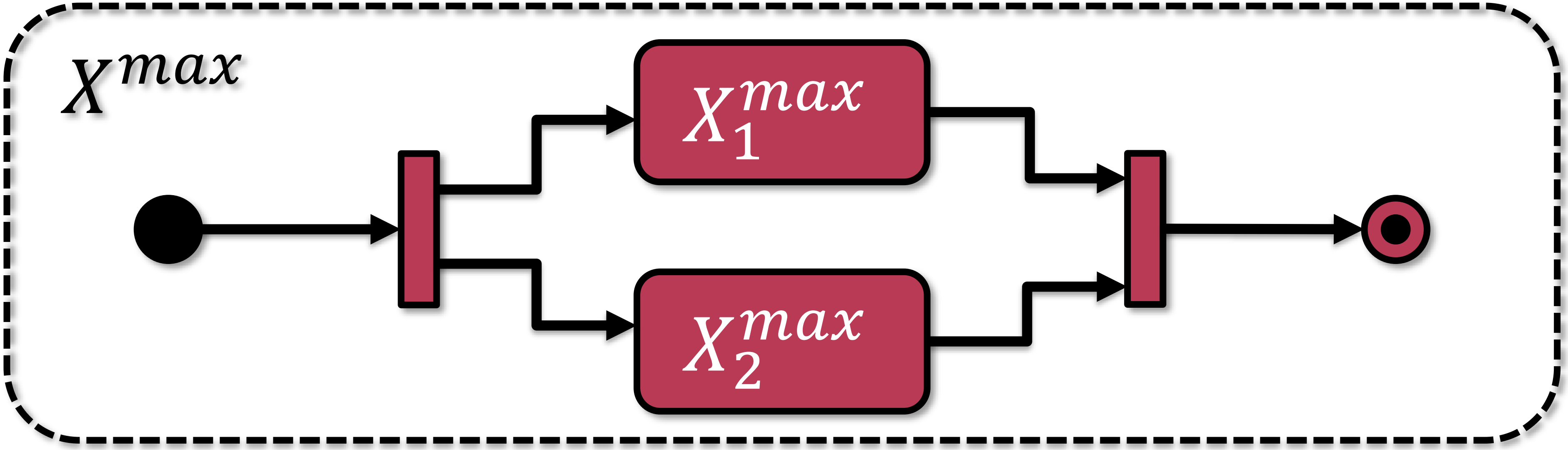



$$X^{max} = min(X_1^{max}, X_2^{max})$$

**Engpass:**

Die Aktivität mit dem minimalen Durchsatz (oder die entsprechende Ressource).

# Parallele Komposition

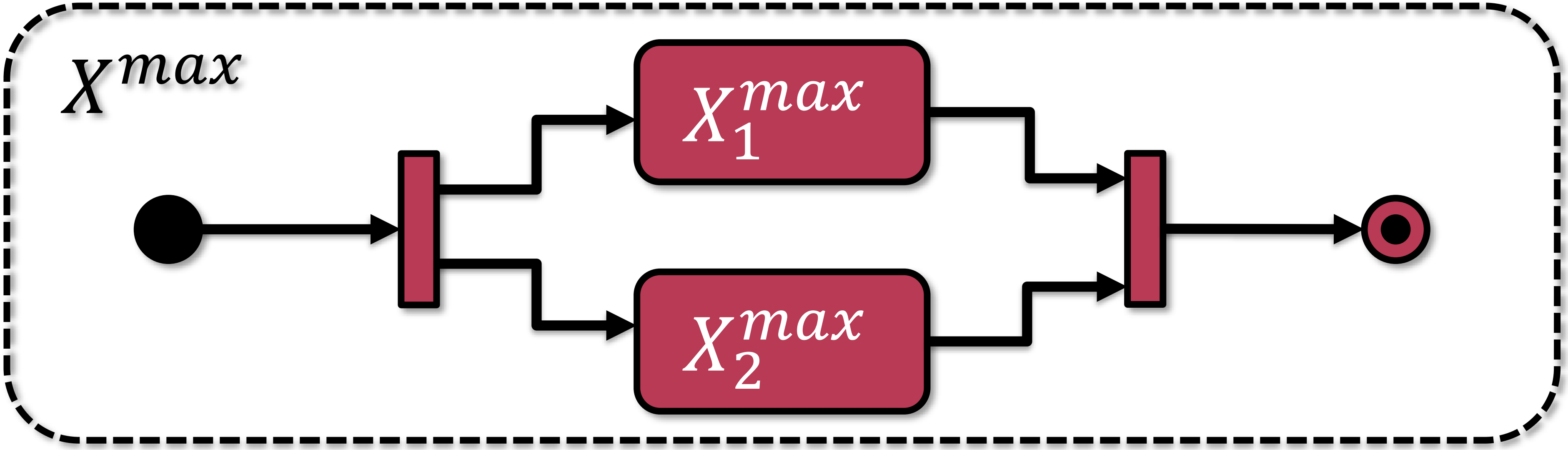



Jede Aktivität wird einmal besichtigt.

Auch wenn die eine Aktivität schnell ausgeführt wird, häufen sich die Tokens vor der anderen an

z.B. Klausurbewertung: Eingangstest (30/Stunde), Großaufgabe (12/Stunde)

# Parallele Komposition

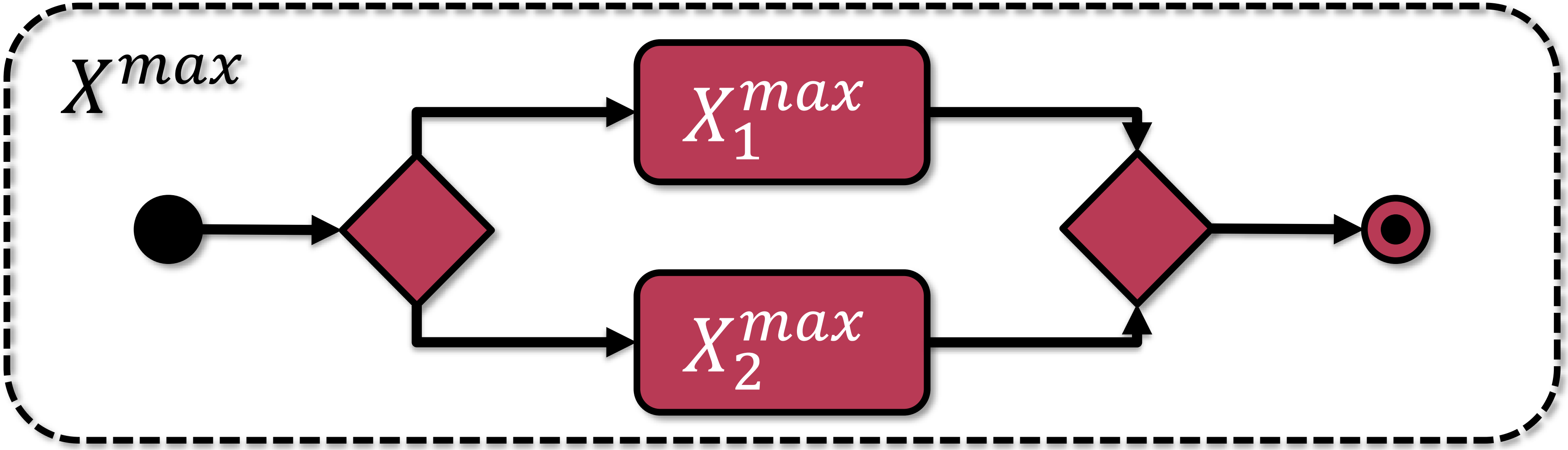



$$X^{max} = min(X_1^{max}, X_2^{max})$$

**Engpass:**

Die Aktivität mit dem minimalen Durchsatz (oder die entsprechende Ressource).

# Freie Wahl

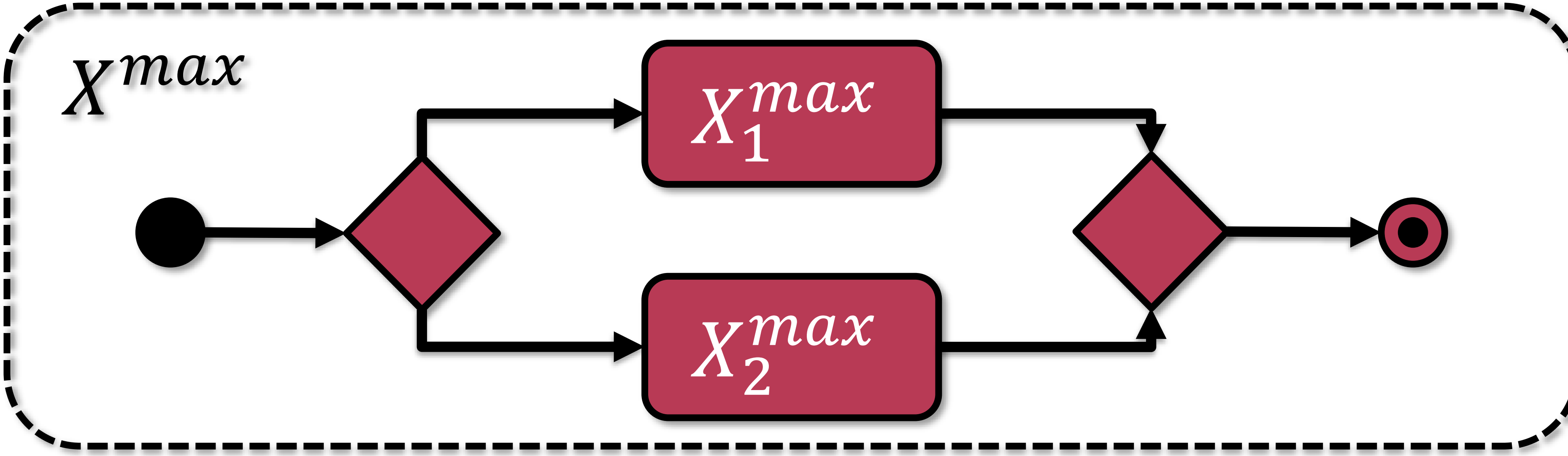



$$X^{max} = X_1^{max} + X_2^{max}$$

Die Tokens können von den zwei Zweigen frei wählen:
falls die eine Aktivität gesättigt wird, die andere kann noch arbeiten.

z.B. in Kaufhaus:
K Kassen, mit je
10 Kunden/Stunde

# Freie Wahl

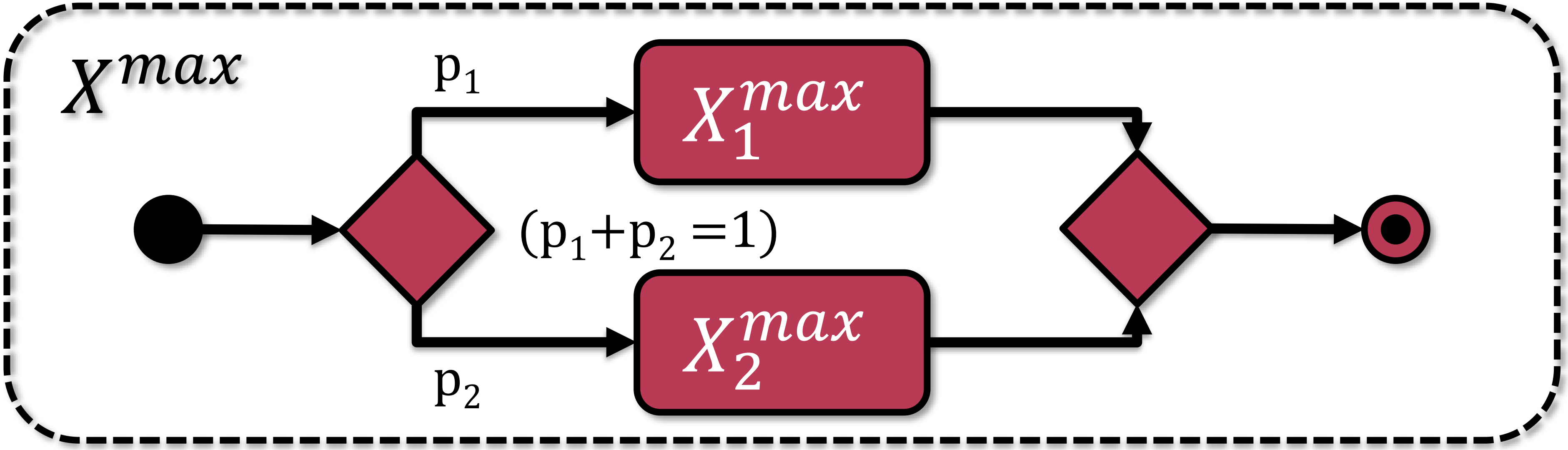



$$X^{max} = X_1^{max} + X_2^{max}$$

Anmerkung: Manchmal wird die Anzahl der Ressourcen bei der Aktivität angegeben, und wird die Aktivität nicht mehrmals graphisch dargestellt. (siehe auch Simulation).

<u>Bedingung</u>: die Ressourcen sollen die Aktivität gleich schnell ausführen können

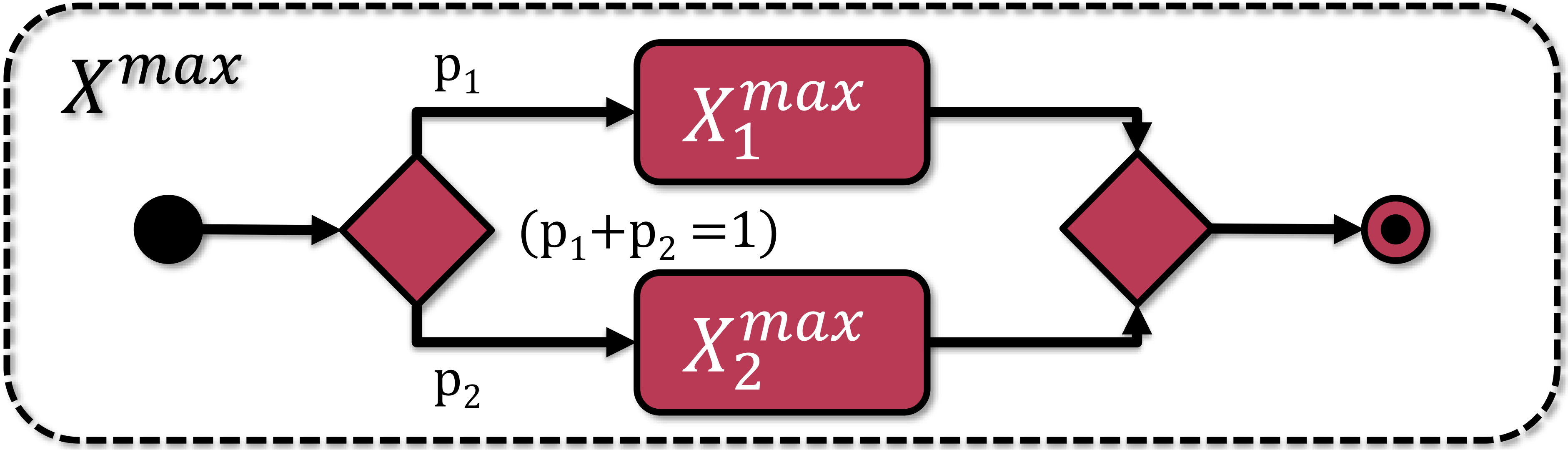
# Wahl mit gegebener Proportion




Die Aktivität $X_1$ wird durchschnittlich $p_1$-mal,
die Aktivität $X_2$ wird durchschnittlich $p_2$-mal besichtigt.

Die Tokens wählen die zwei Zweige mit den Wahrscheinlichkeiten $\boldsymbol{p_1}$ und $\boldsymbol{p_2}$. Also ein Token aus $\frac{1}{p_1}$ bzw. $\frac{1}{p_2}$ wählt die erste bzw. zweite Aktivität.

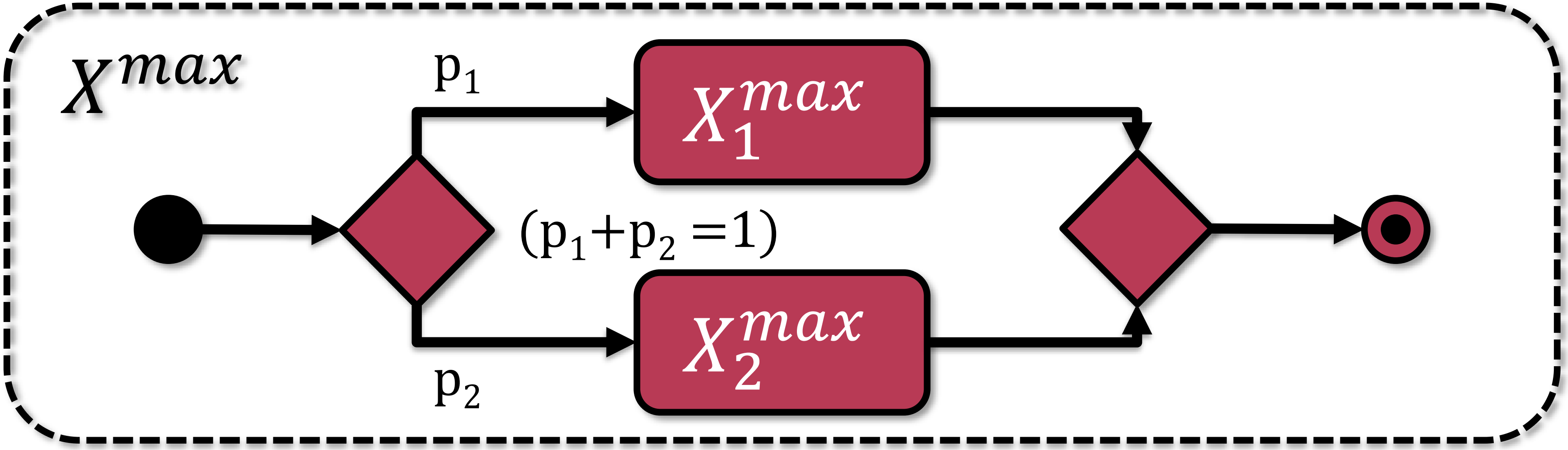
# Wahl mit gegebener Proportion




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

z.B. das Verhalten der Kunden an einer Webseite:
mit 20% Wahrscheinlichkeit kauft er (20/s),
mit 80% bricht er ab (200/s)

# Wahl mit gegebener Proportion

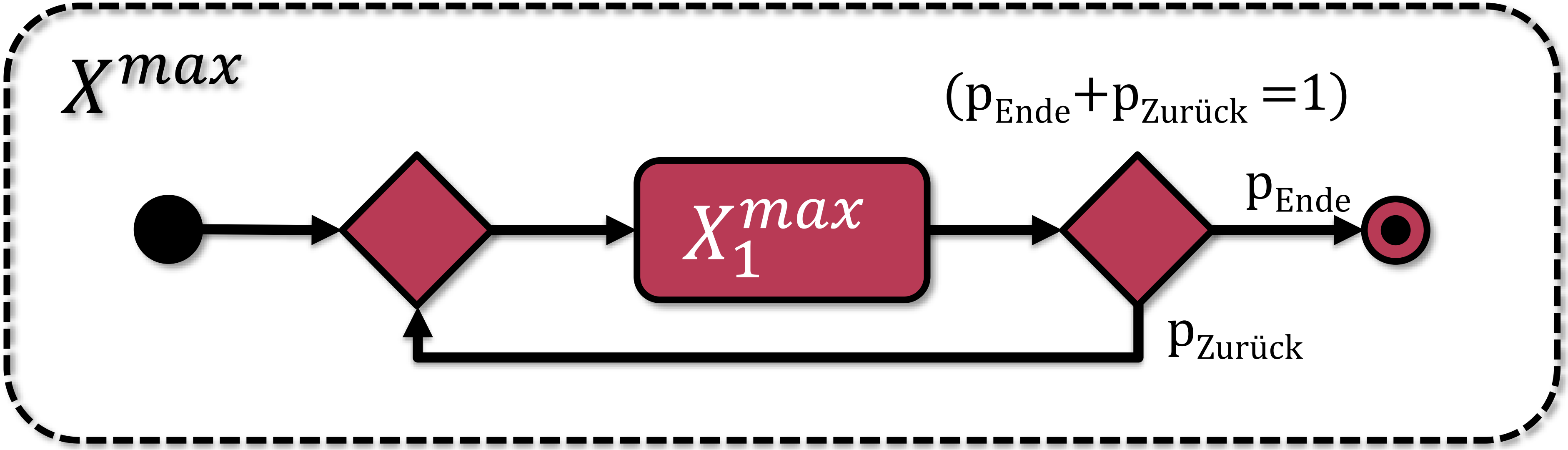



$$X^{max} = min\left(\frac{1}{p_1} \times X_1^{max}, \frac{1}{p_2} \times X_2^{max}\right)$$

**Engpass:**

Die Aktivität mit dem minimalen relativen Durchsatz (oder die entsprechende Ressource).

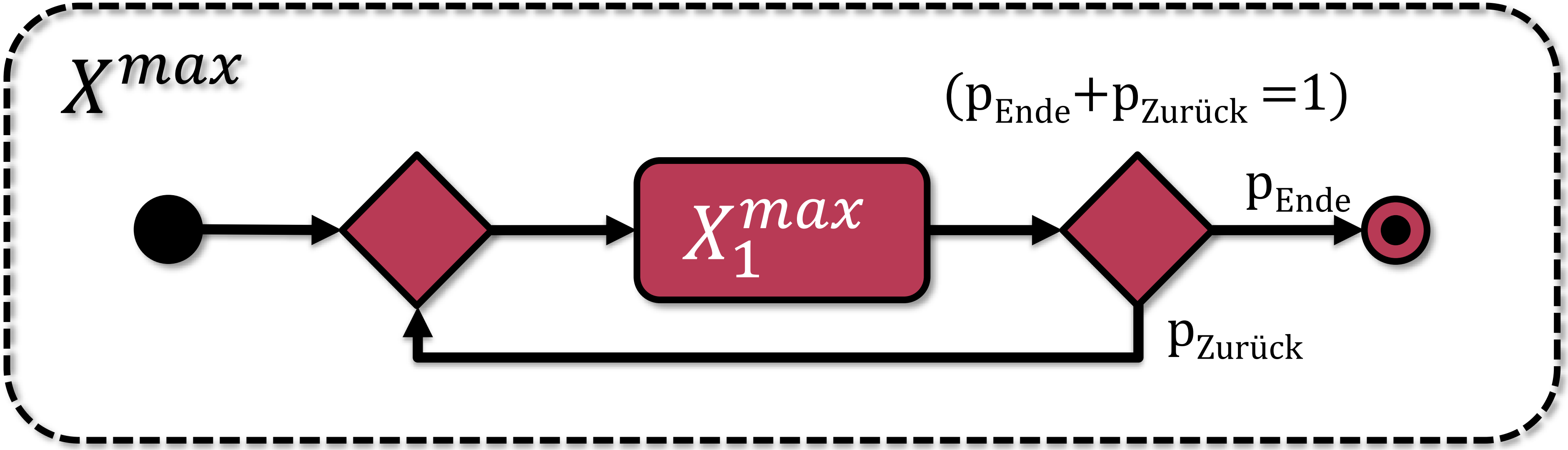
# Komposition mit Schleife




Die Aktivität $X_1$ wird durchschnittlich $\frac{1}{p_{Ende}}$-mal, besichtigt.

(Wenn $p_{Ende}$ ⅓ ist, dann 3-mal, wenn ⅕, dann 5-mal. Siehe Wahrscheinlichkeitsrechnung.)

# Komposition mit Schleife




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

z.B. der Kunde in einem System:
mit 10% bricht er ab, mit 90% stellt neue Frage
15 Fragen/s, durchschnittlich 10 Fragen/Sitzung

# Definition: Besichtigungszahl

Die **Besichtigungszahl** gibt an, wie oft die gegebene Aktivität/Unterprozess während der Ausführung abläuft.

- Wie oft wird der Prozess während einer Ausführung eine gegebene Aktivität besichtigen? (Visitationen)
- Während einer Ausführung eines Prozesses kann eine Aktivität gar nicht, einmal oder mehrmal ablaufen (siehe Verzweigungen, Schleifen).
  - Wenn die mögliche Wahl zwischen verschiedenen Ausgängen mit **Wahrscheinlichkeiten** beschrieben ist, dann spielen diese Wahrscheinlichkeiten bei der Berechnung der Besichtigungszahl eine wichtige Rolle.

# Besichtigungszahl

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

Schleife: $$X^{max} = \frac{1}{\frac{1}{p_{Ende}}} \times X_1^{max} = p_{Ende} \times X_1^{max}$$

- Die **Besichtigungszahl** gibt an, wie oft die gegebene Aktivität/Unterprozess während der Ausführung abläuft.
  - Bei einer Wahl ist sie die Wahlwahrscheinlichkeit
  - Bei einer Schleife ist sie die erwartete Iterationsanzahl

# Besichtigungszahl

Grenzdurchsatz bei gegebener Besichtigungszahl:

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

- **Besichtigungszahl:** gibt an, wie oft die gegebene Aktivität/Unterprozess während der Ausführung abläuft.
  - Bei einer Wahl ist sie die Wahlwahrscheinlichkeit
  - Bei einer Schleife ist sie die erwartete Iterationsanzahl

# Besichtigungszahl

Grenzdurchsatz bei gegebener Besichtigungszahl:

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

- **Besichtigungszahl:** gibt an, wie oft die gegebene Aktivität/Unterprozess während der Ausführung abläuft.
  - Bei einer Wahl ist sie die Wahlwahrscheinlichkeit
  - Bei einer Schleife ist sie die erwartete Iterationsanzahl

# Besichtigungszahl

Abfertigungszeit bei gegebener Besichtigungszahl:

$$T_{Prozess} = v \times T_{Task}$$

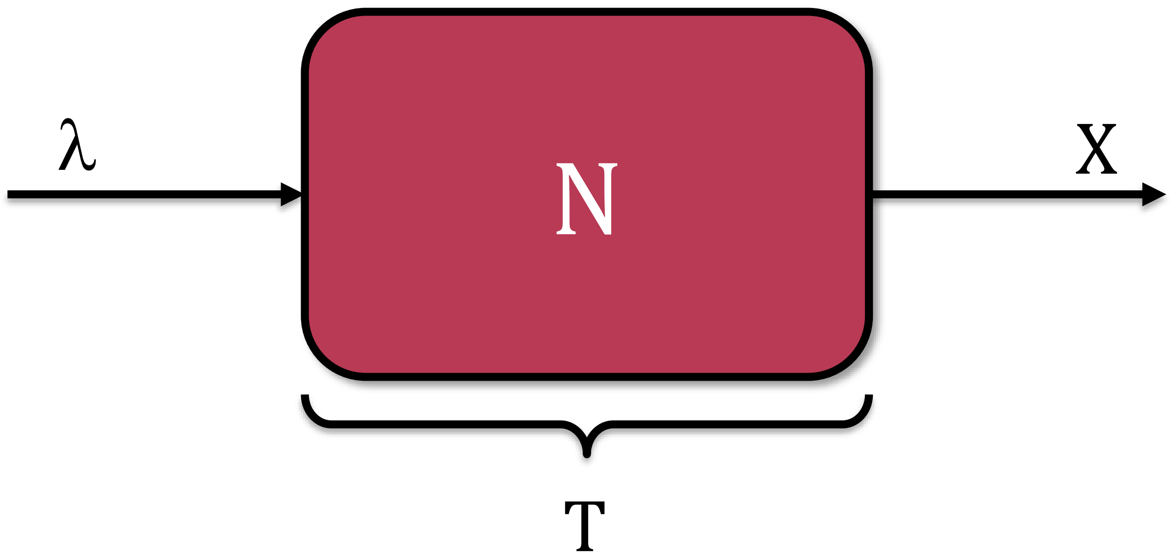
- **Besichtigungszahl:** gibt an, wie oft die gegebene Aktivität/Unterprozess während der Ausführung abläuft.
  - Bei einer Wahl ist sie die Wahlwahrscheinlichkeit
  - Bei einer Schleife ist sie die erwartete Iterationsanzahl

# DER SATZ VON LITTLE

Die Grundformel

# Der Satz von Little

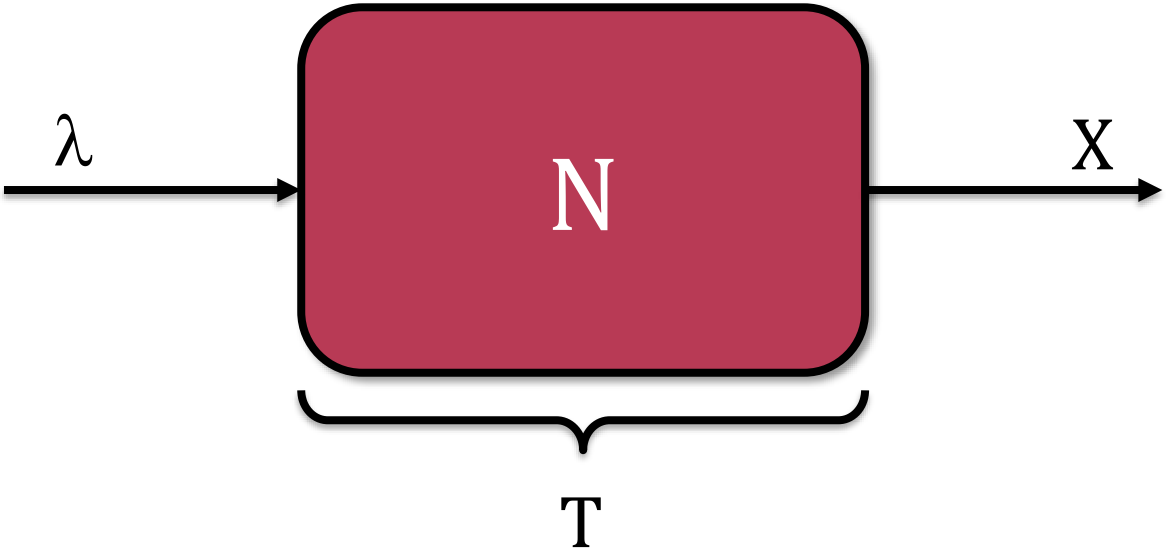
- λ: Ankunftsrate $\left[\frac{Anfrage}{Sec}\right]$
- X: Durchsatz $\left[\frac{Anfrage}{Sec}\right]$
- T: Abfertigungszeit $[Sec]$
- N: Anzahl der Token im System $[Anfrage]$



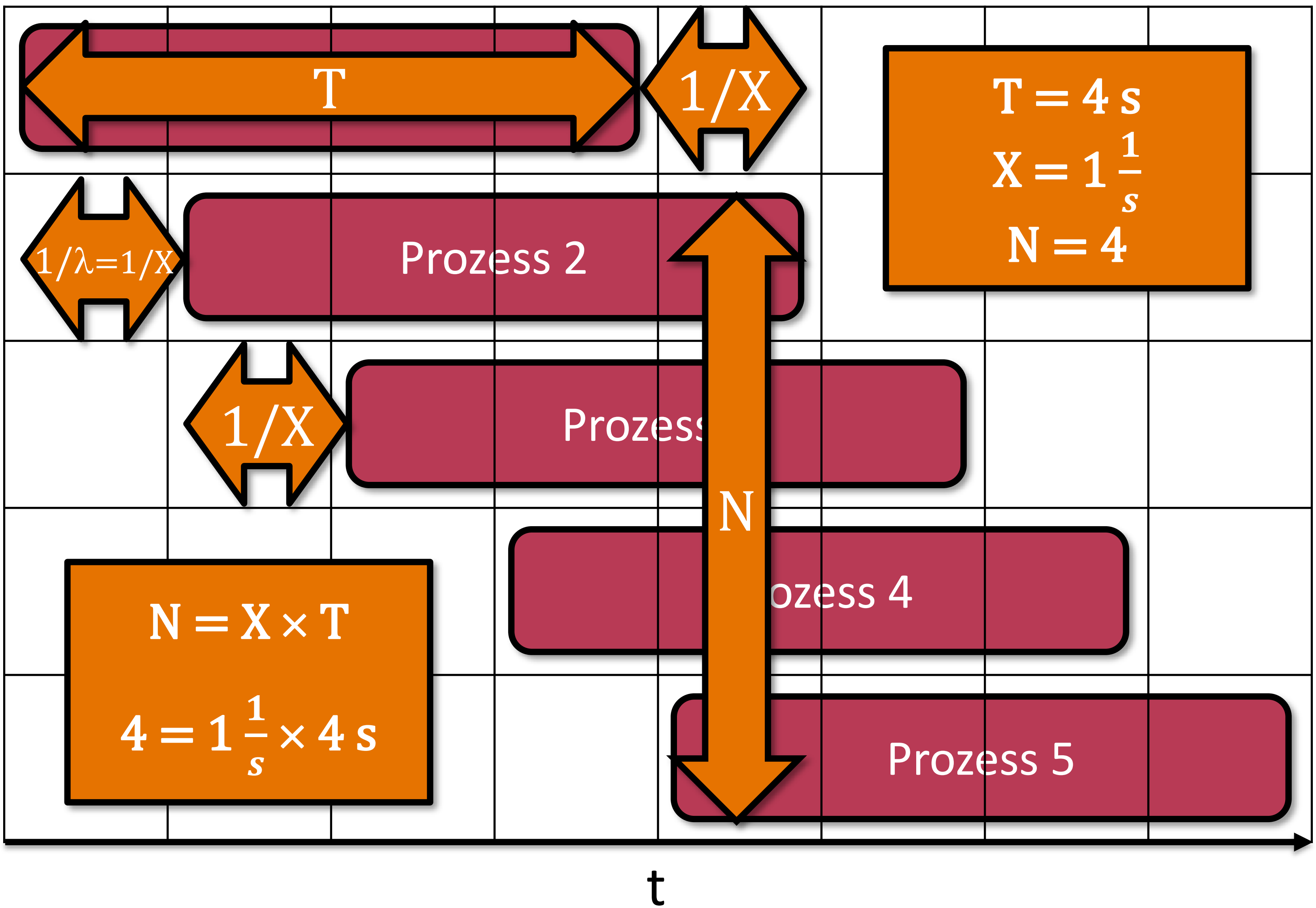
# Der Satz von Little

**Im stabilen Zustand (im Gleichgewicht, $\lambda$ = X)**
gilt der Satz von Little:

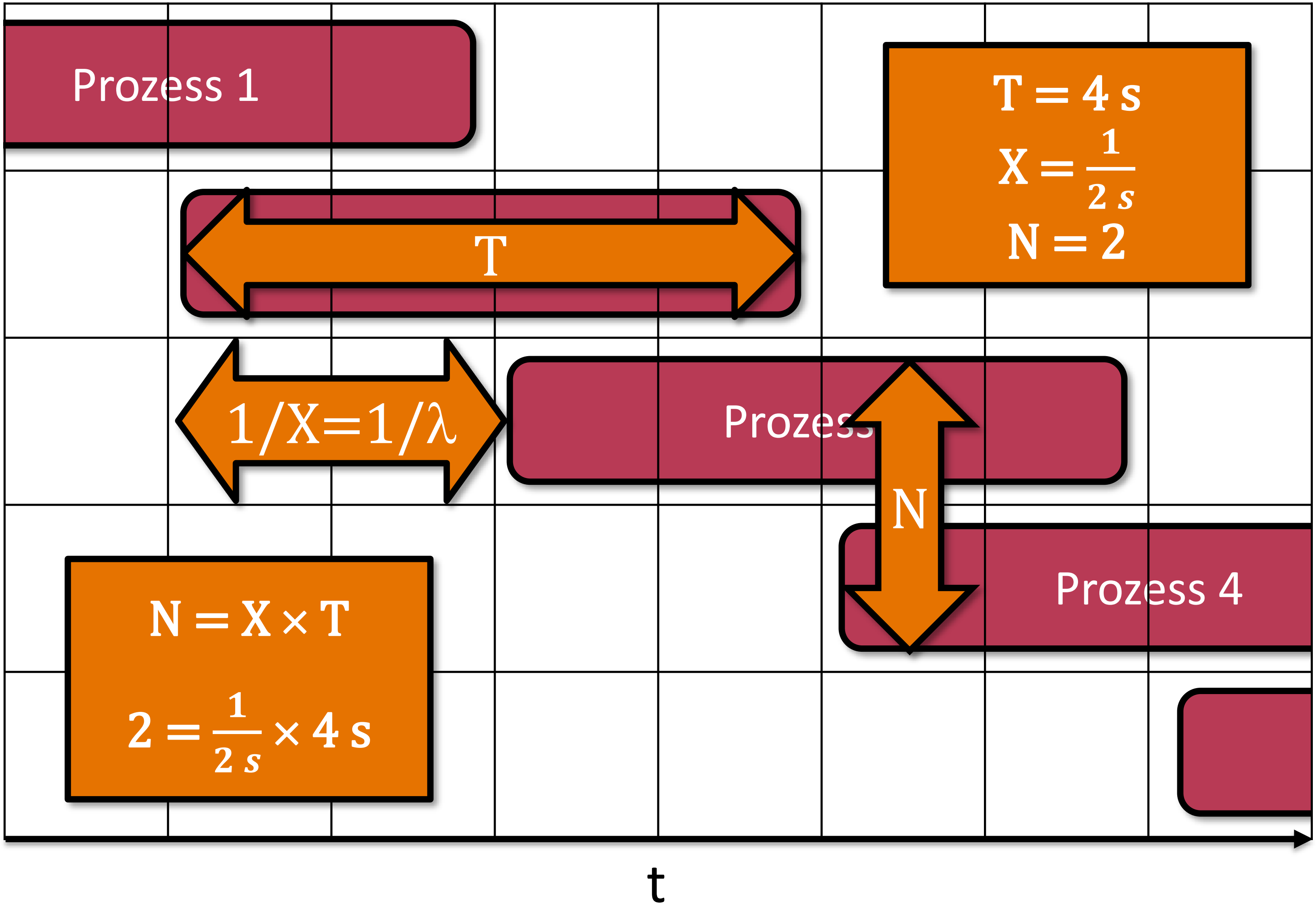
$$N = X \times T$$



# Illustration des Satzes von Little



# Illustration des Satzes von Little



# Die Auslastung und der Satz von Little

- K Serverinstanzen: maximal K Prozessinstanzen können gleichzeitig unter Abfertigung stehen
- Der Satz von Little gibt die Anzahl der unter Abfertigung stehender Prozessinstanzen (N) an
- Davon ist folgendes ableitbar:

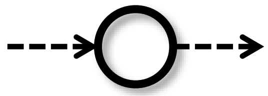
$$\boldsymbol{U} = \frac{\boldsymbol{X}}{\boldsymbol{K}} \times \boldsymbol{T} = \frac{\boldsymbol{X} \times \boldsymbol{T}}{\boldsymbol{K}} = \frac{\boldsymbol{N}}{\boldsymbol{K}}$$

Auslastung bei K Serverinstanzen

Der Satz von Little ($N = X \times T$)

# DER SATZ VON LITTLE: PRAKTISCHE BEISPIELE

# Der Satz von Little – Beispiel



Festplatte

- Ressource/Bedienungseinheit: Festplatte
- Sie bedient 40 Anfragen/Sekunden (keine Überlappung)
- Die Bedienzeit einer Anfrage beträgt durchschnittlich 0,0225 Sekunden
- Wie hoch ist die Auslastung?

$$U = X \times T_{Festplatte} = 40\,\frac{Anfrage}{Sec} \times 0{,}0225\ Sec = 0{,}9 = 90\%$$

# Der Satz von Little – Beispiel


- Die Anfragen werden in eine Warteschlange vor der Festplatte gestellt
- Durchsatz der Festplatte: 40 $Anfragen/Sek$
- Die durchschnittliche Anzahl der Anfragen im System: 4

Durchschnittliche Antwortzeit? ($T_{System}$)
Die Zeit, die die Anfrage im System verbringt?

Durchschnittliche Wartezeit? ($T_{Warten}$)
Die Zeit, die die Anfrage in der Warteschlange verbringt?

# Der Satz von Little – Beispiel


- Die Anfragen werden in eine Warteschlange vor der Festplatte gestellt
- Durchsatz der Festplatte: 40 $Anfragen/S$
- Die durchschnittliche Anzahl der Anfragen im …em: 4

**Wartezeit und Abfertigungszeit zusammen**

Durchschnittliche Antwortzeit? ($T_{System}$)

$$N = X \times T \rightarrow T_{System} = 4\ Anfragen/\ 40\ \frac{Anfr.}{Sek} = 0{,}1\ Sek$$

Durchschnittliche Wartezeit? ($T_{Warten}$)

$$(T_{System} - T_{Festplatte}) = (0{,}1\ s - 0{,}0225\ s) = 0{,}0775\ s$$

# Der Satz von Little – Beispiel


- Die Anfragen werden in eine Warteschlange vor der Festplatte gestellt
- Durchsatz der Festplatte: 40 $Anfragen/Sek$
- Die durchschnittliche Anzahl der Anfragen im System: 4

Durchschnittliche Anzahl der Anfragen in der Warteschlange?

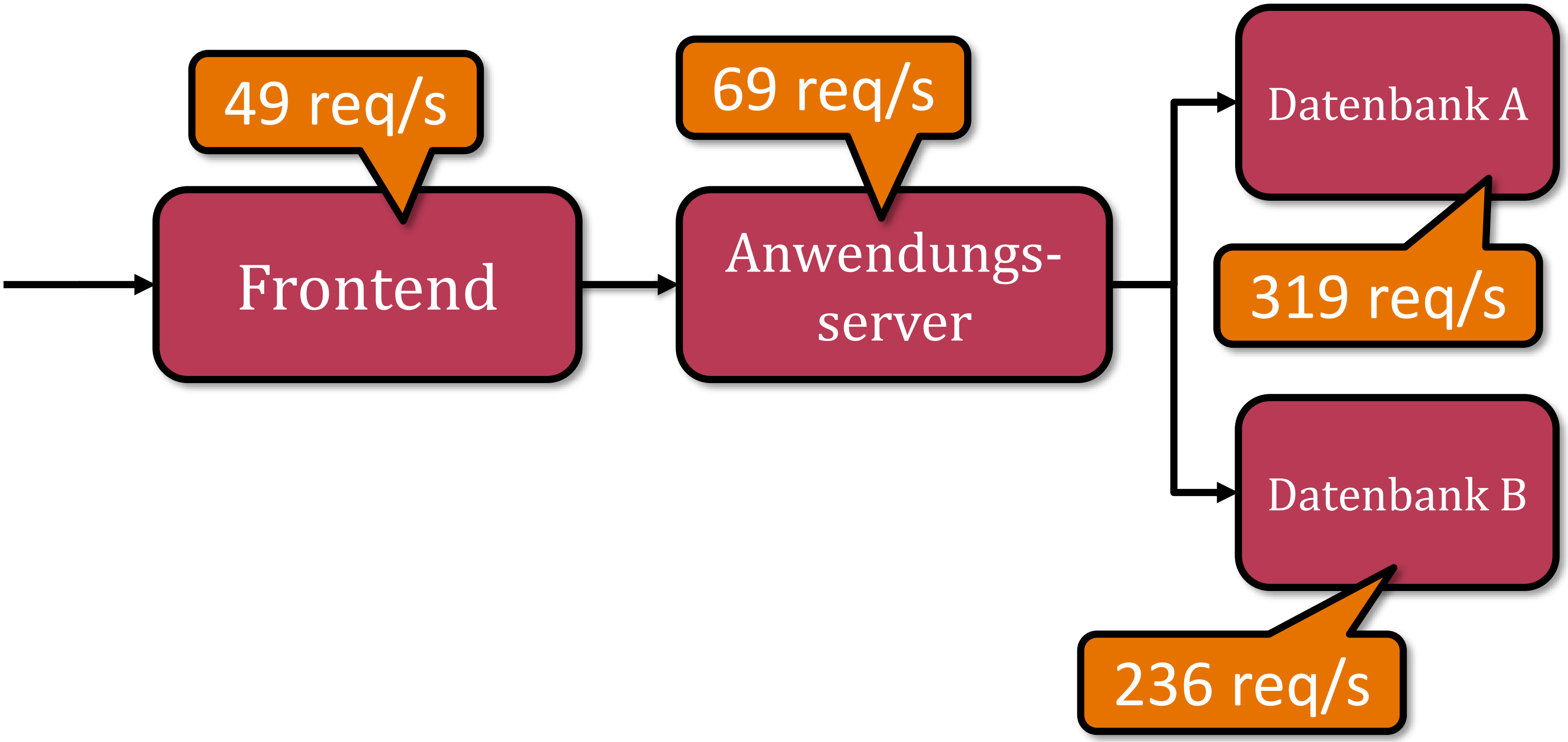
$(N_{System} - N_{Festplatte})$

4 $Anfragen$ – 0,9 $Anfragen$ = 3,1 $Anfragen$

# Der Satz von Little in der Praxis

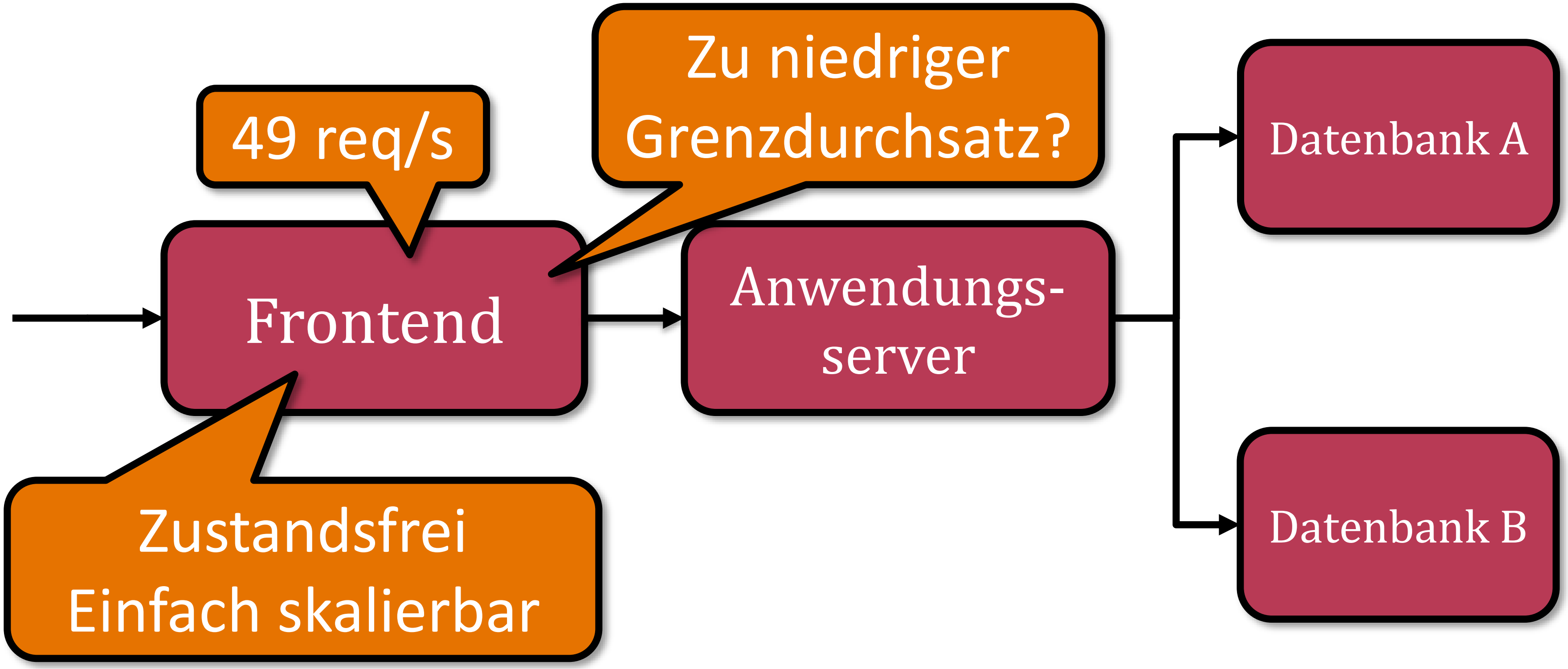
- Simulation
  - Dobson&Shumsky
  - https://youtu.be/UjzXQPGBaNA
- Warum wird es unterrichtet?
  - http://pubsonline.informs.org/doi/pdf/10.1287/ited.7.1.106
- Beispiele
  - http://web.mit.edu/sgraves/www/papers/Little's%20Law-Published.pdf
    - Z.B.: Wie lange liegen die Weinflaschen im Keller?
      - Der Keller ist durchschnittlich so um ⅔ voll. (~160 Flaschen)
      - Wir kaufen monatlich durchschnittlich um die 8 Flaschen.
      - Laut dem Satz von Little liegen die Flaschen $T=N/X$, also $160/8=20$ Monate im Keller.

# Leistung in einem 3-Schichten-System



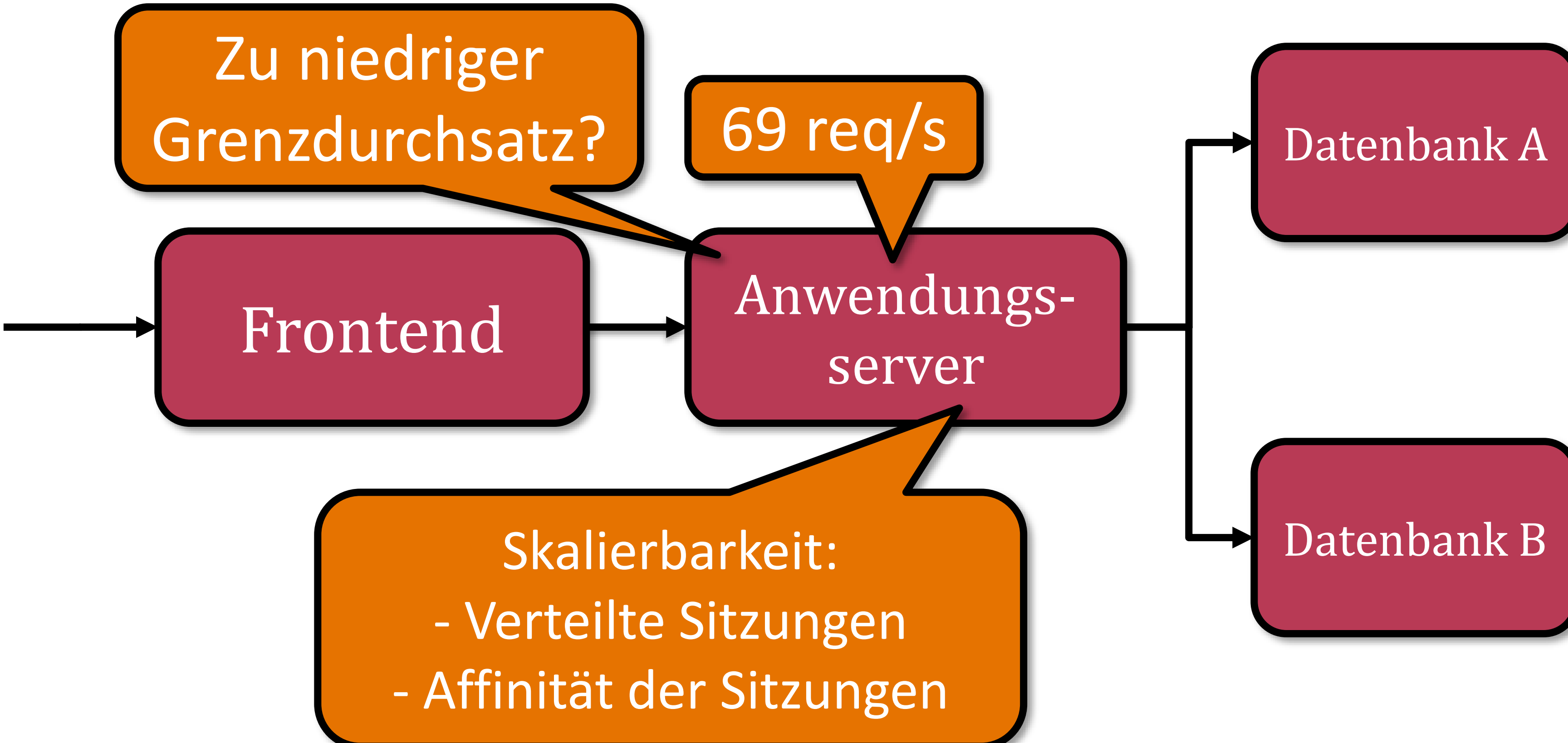

Diese Zahlen beziehen sich auf die Ankunftsrate des Gesamtsystems! Z.B. „Datenbank A” wird zum Engpass, wenn das Gesamtsystem 319 Anfragen pro Sekunde bekommt.

# Leistung in einem 3-Schichten-System



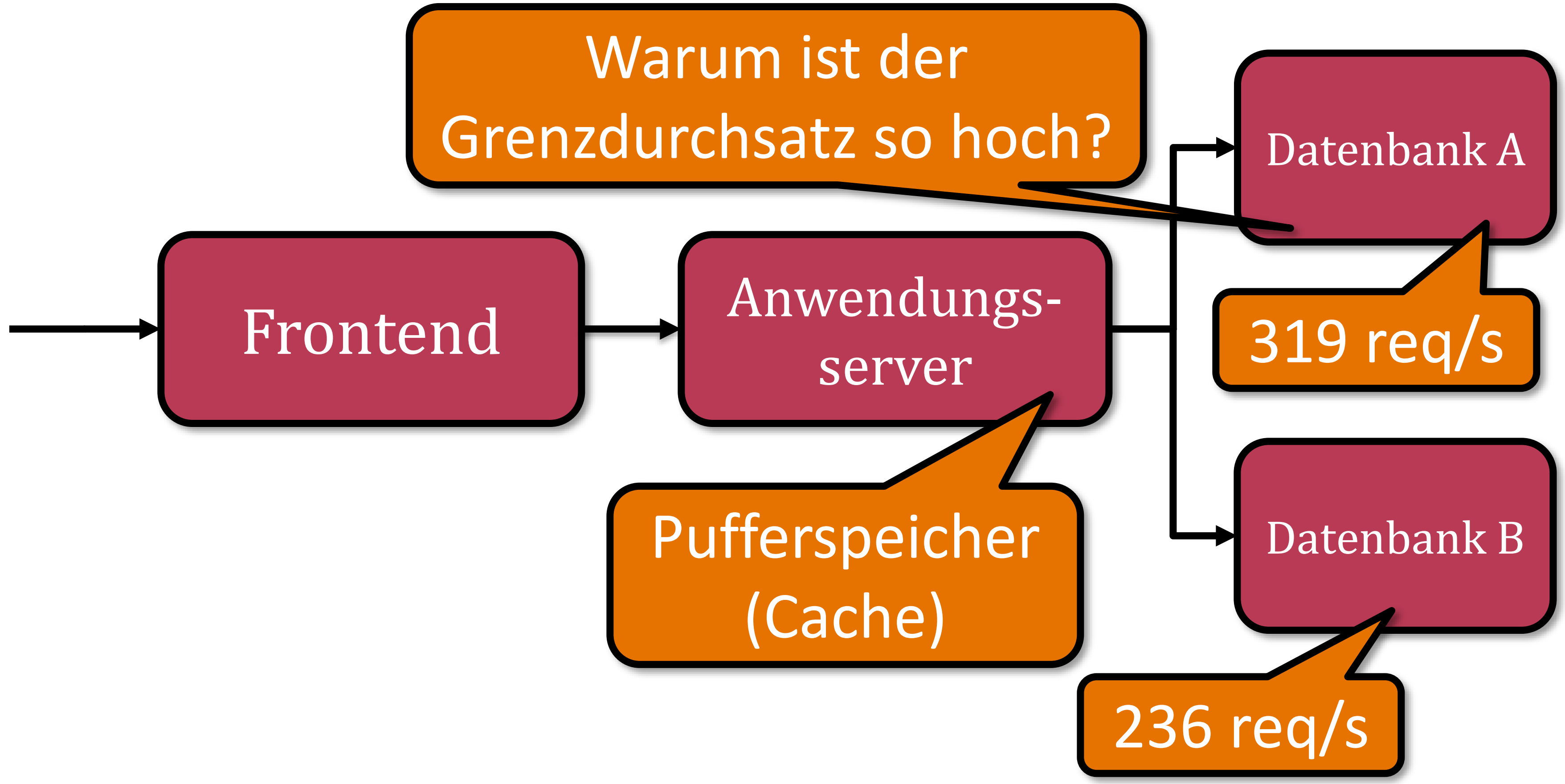

Diese Zahlen beziehen sich auf die Ankunftsrate des Gesamtsystems! Z.B. „Datenbank A" wird zum Engpass, wenn das Gesamtsystem 319 Anfragen pro Sekunde bekommt.

# Leistung in einem 3-Schichten-System



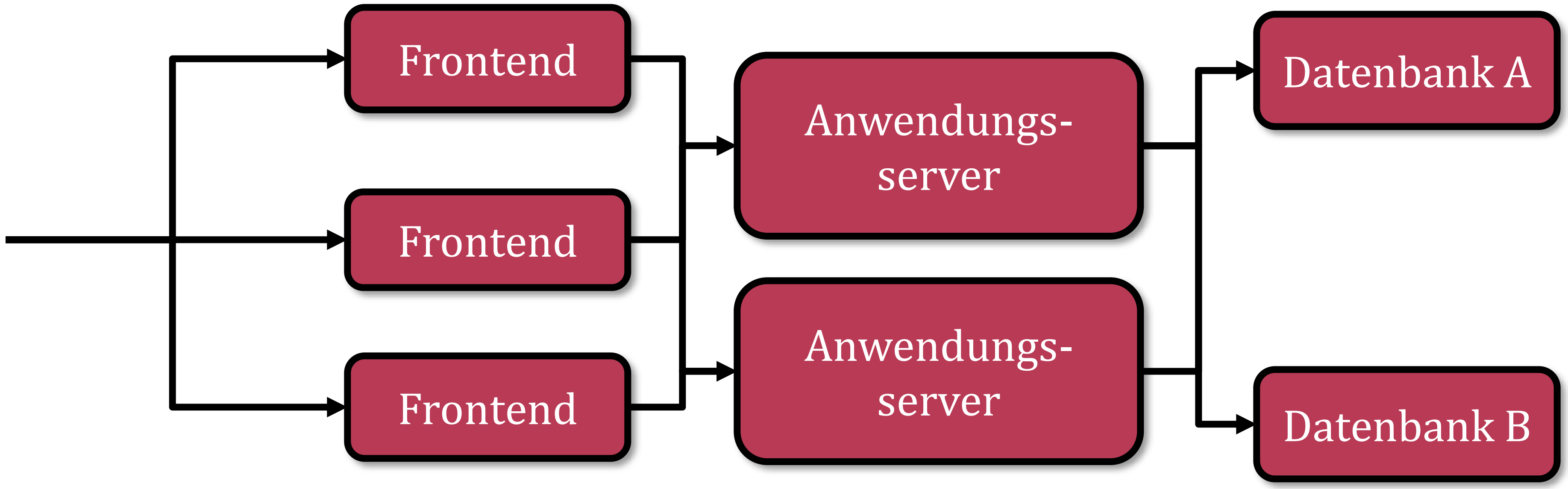

Diese Zahlen beziehen sich auf die Ankunftsrate des Gesamtsystems! Z.B. „Datenbank A” wird zum Engpass, wenn das Gesamtsystem 319 Anfragen pro Sekunde bekommt.

# Leistung in einem 3-Schichten-System




Diese Zahlen beziehen sich auf die Ankunftsrate des Gesamtsystems! Z.B. „Datenbank A” wird zum Engpass, wenn das Gesamtsystem 319 Anfragen pro Sekunde bekommt.

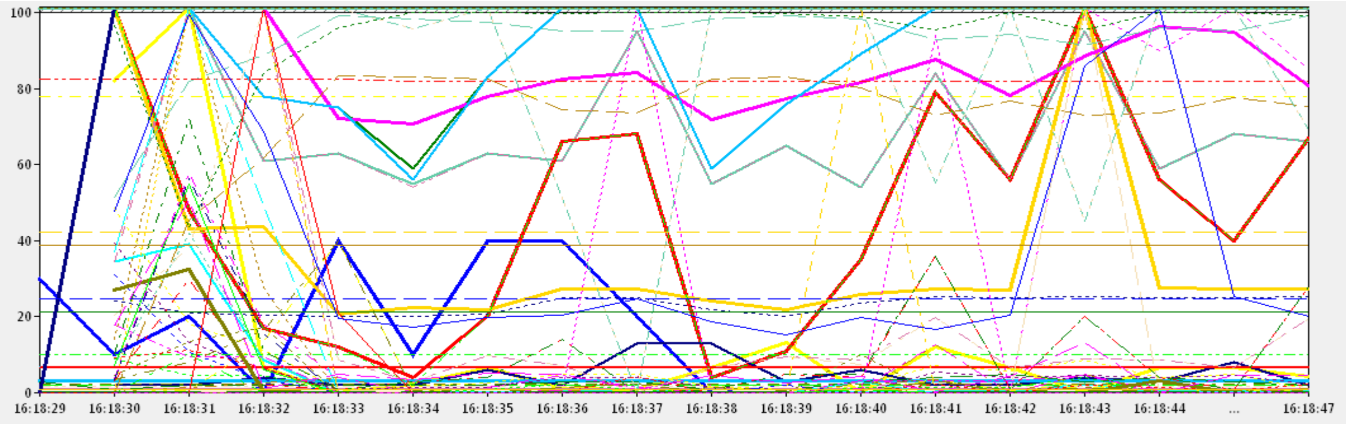
# 3-Schichten-Architektur in der Wirklichkeit




(Beispiel für den Interessierten:)
http://www.projectclearwater.org/wp-content/uploads/2013/05/Clearwater-Deployment-Sizing-10-Apr-13.xlsx
http://www.projectclearwater.org/technical/clearwater-performance/

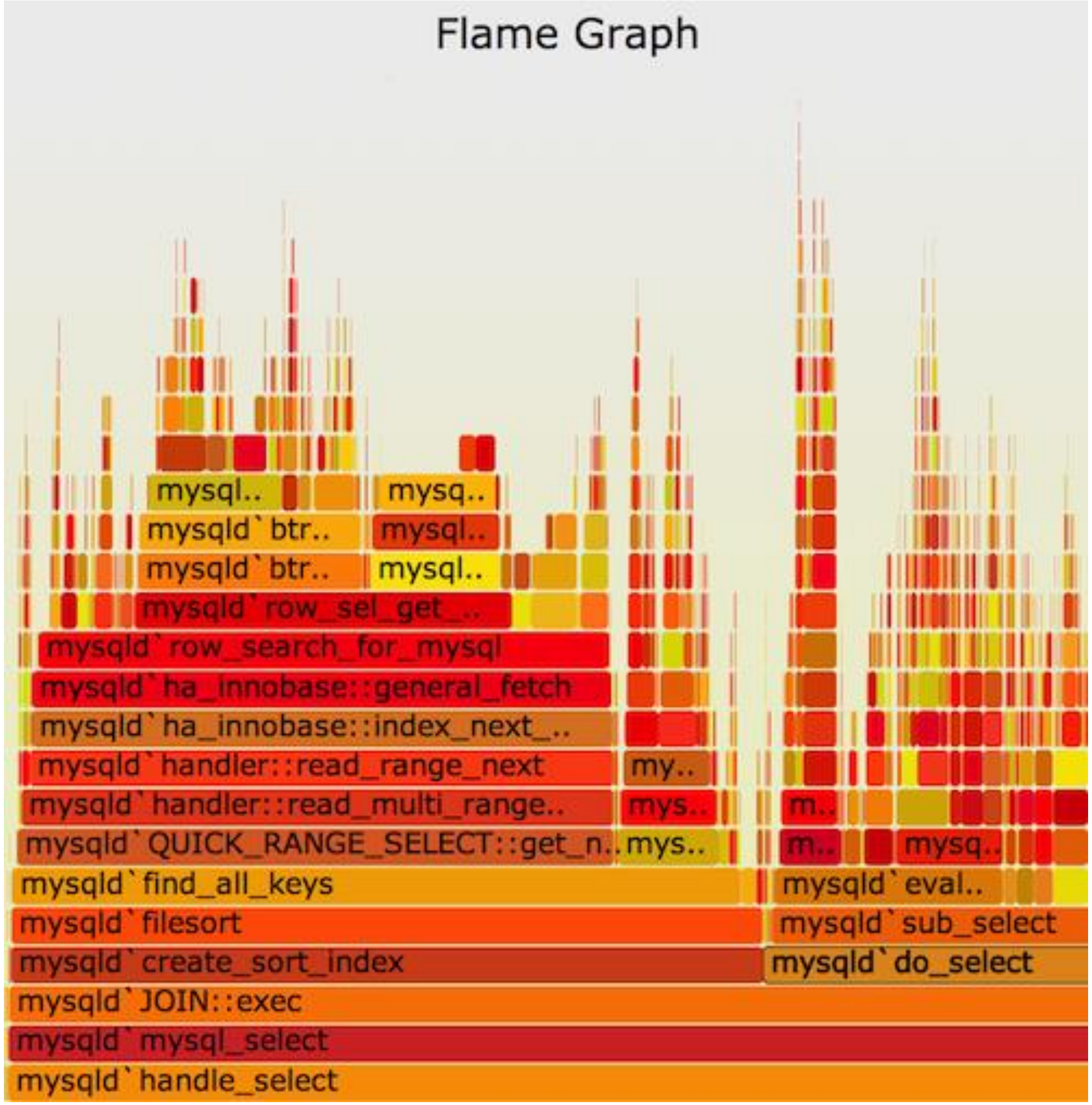
# Was soll gemessen werden?

- Was ist wichtig?
- Metriken „im kleinen”
  - z.B. Task manager, Resource monitor, das gleiche auf Serverseite




- Metriken “im großen”
  - z.B. virtualisierte Infrastrukturen
- Was ist wichtig denn?

# Beispiel: was wird so lange gerechnet ….?



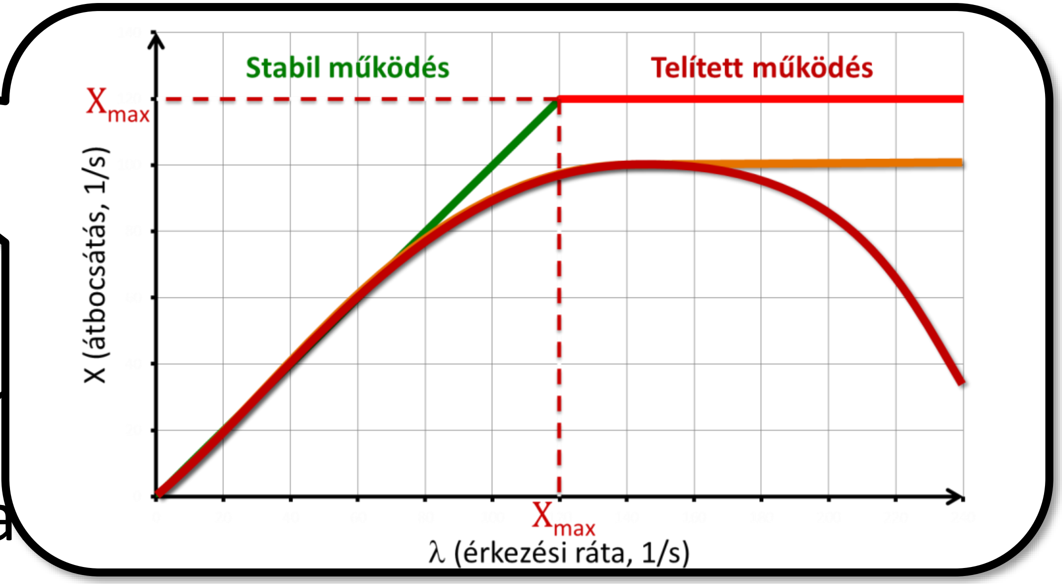

http://www.brendangregg.com/flamegraphs.html

# Lug und Trug

- In der Praxis sind die Werte nicht einfach zu messen
  - z.B. die Antwortzeit variiert
  - die Ankunftsrate variiert auch
- Die Anwendungen konkurrieren
  - (2*X != X + X)
- Die Bedienzeit einer Aufgabe kann Datenabhängig sein
- Die Ressourcen sollen richtig ausgewählt werden
  - → Die Lastverteilung kann kritisch sein
- Ankunftsorder/-muster kann vernachlässigt werden!
  - Dies ist die Stärke des Satzes von Little
- Struktur/Parameter des Systems kann auch variabel sein

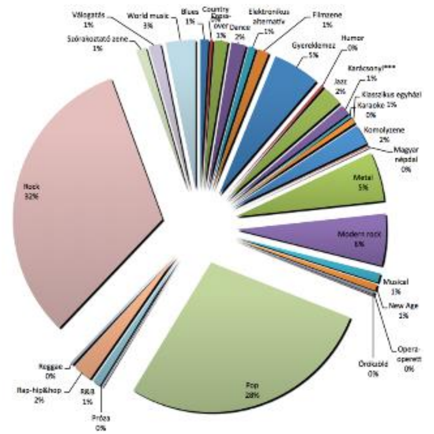
# Lug und Trug

- In der Praxis sind die Werte nicht einfach zu messen
  - z.B. die Antwortzeit variiert
  - Die Ankunftsrate variiert auch
- Die Anwendungen konkurrieren
  - (2*X != X + X)
- Die Bedienzeit eine
  Datenabhängig sein
- Die Ressourcen soller
  → Die Lastverteilung ka



# LASTMODELLE: DER SATZ VON ZIPF

# Was ist der Inhalt der Anfragen?



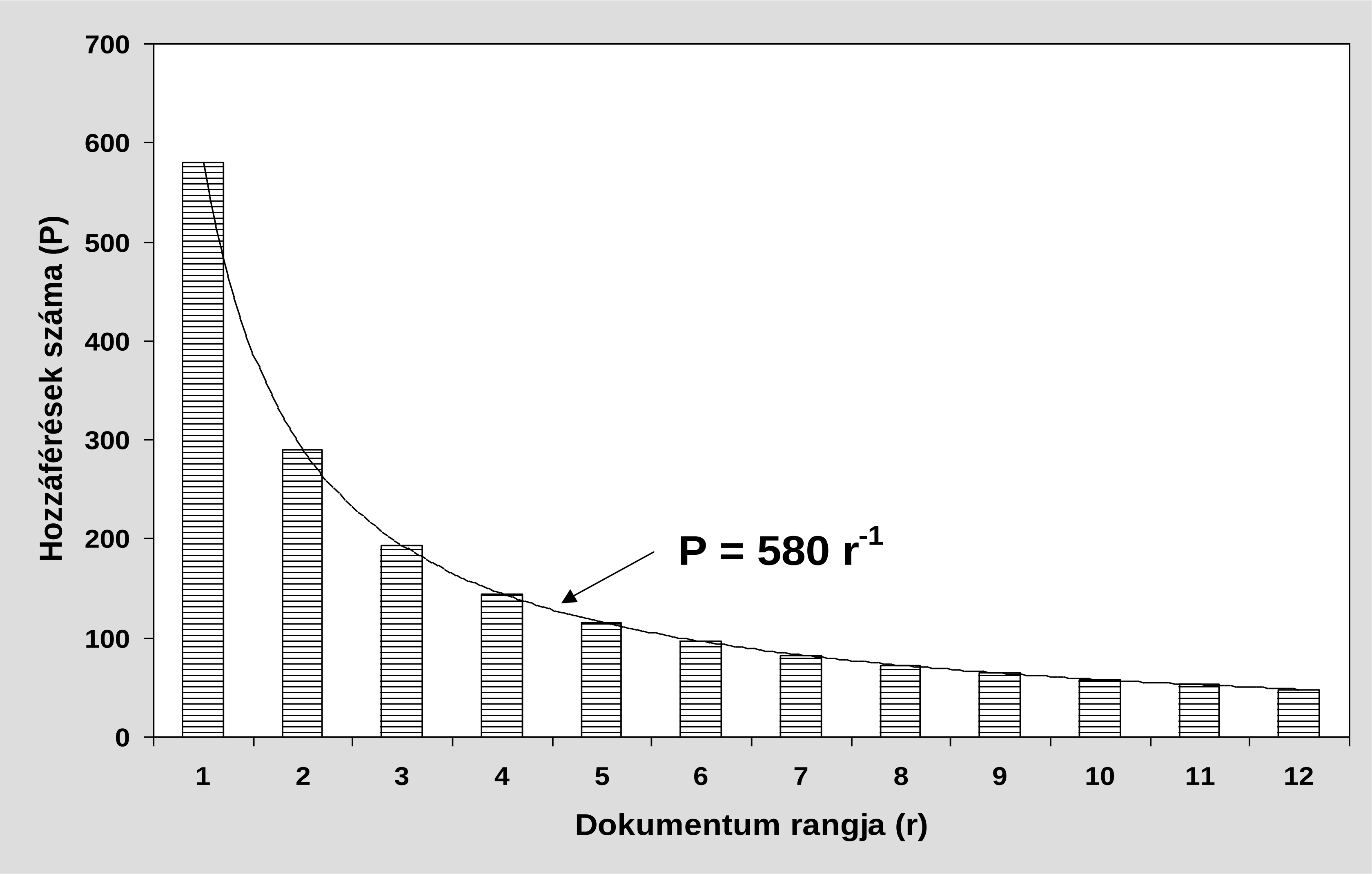

- Bisher: die Anfragen sind gleich
  - „Anfrage der Daten eines Buches”
- Eigentlich: die Anfragen haben Inhalt
  - „Anfrage der Daten des *Joseph und seine Brüder*”
  - siehe Pareto-Prinzip (Operationssysteme, Datenbanken, …)
  - Die (Mehrheit der) Anfragen beziehen sich auf ein (kleines) Teil der Daten (80% – 20%)
- Wichtig, weil …
  - Technische Auswirkungen
    - Cache, pool size, statischer Speicher, …
  - Betrifft das Systemmodell auch
    - Häufige Anfragen werden getrennt behandelt

# Der Satz von Zipf

- Ursprünglich: Auftrittshäufigkeit von Wörter in *Korpustexten* weist eine markante Verteilung auf
  - Es stellte sich heraus, dass es nicht nur für sprachgebundene Texte gilt
  - Anwendungen in mehreren Wissenschaftsgebieten

George Kingsley Zipf
(1902–1950)
US-amerikanischer Linguist

# Der Satz von Zipf – Beispiele



# Der Satz von Zipf – Beispiele

- Hitlisten
- Einwohneranzahl von Städten nach ihrer Rangfolge
- Charakteristik des Internet-Verkehrs
- Beliebtheit von Unterseiten von Webseiten
- Die Entwicklung der open source OS

- (im Allgemeinen: Potenzgesetz)

# Der Satz von Zipf – Die Formel

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

- $R_i$ – Häufigkeit des i. Wortes
  - i=1 für das häufigste Wort
  - i=2 für das zweithäufigste
  - …
- α – korpusspezifischer Wert
  - in der Nähe von 1
- Vereinfachung (α = 1):
  - f (frequency): Auftrittshäufigkeit
  - p (popularity): Rang des Textes (in fallender Reihenfolge)

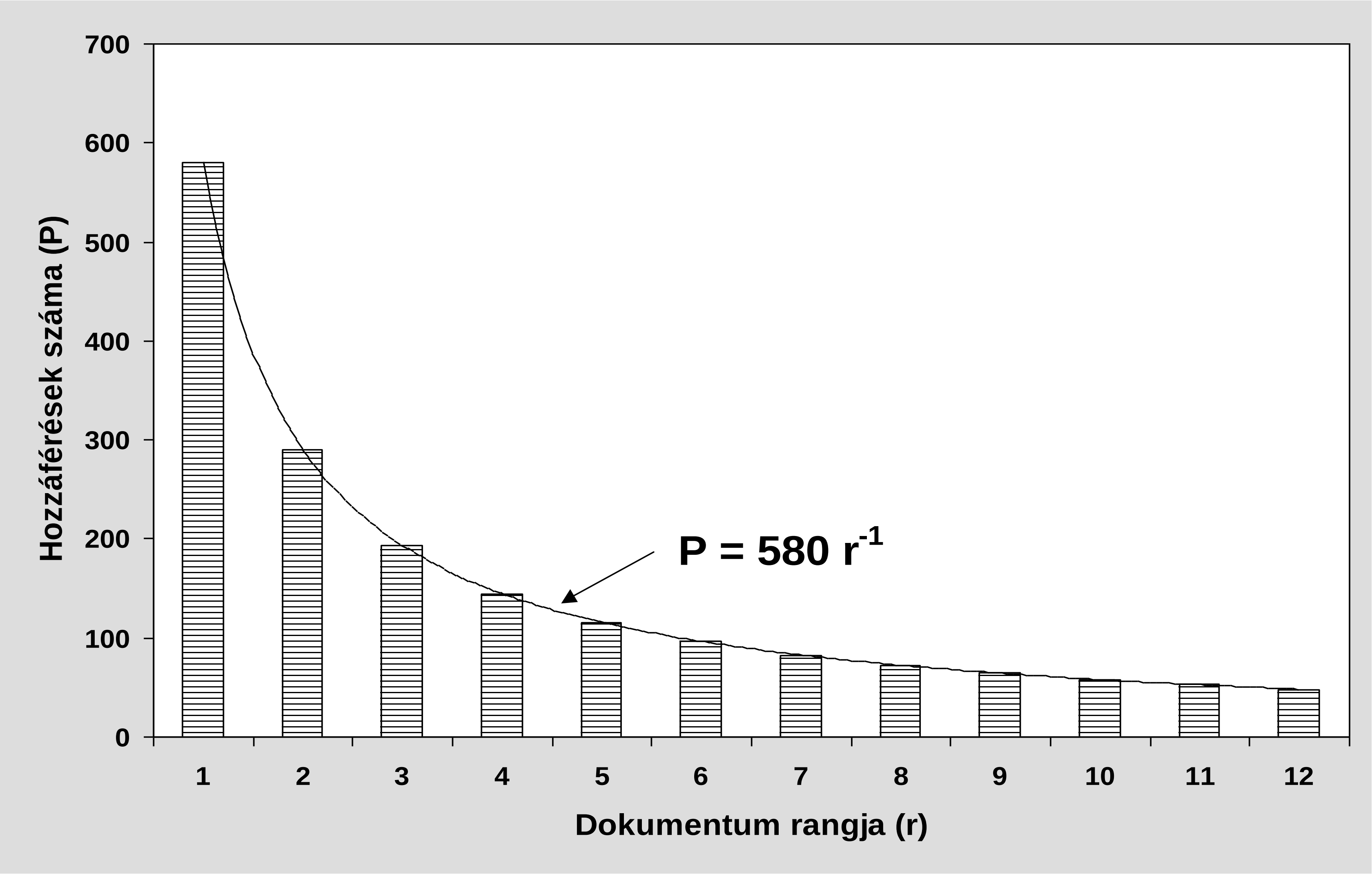
# Der Satz von Zipf – für Webdokumente

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

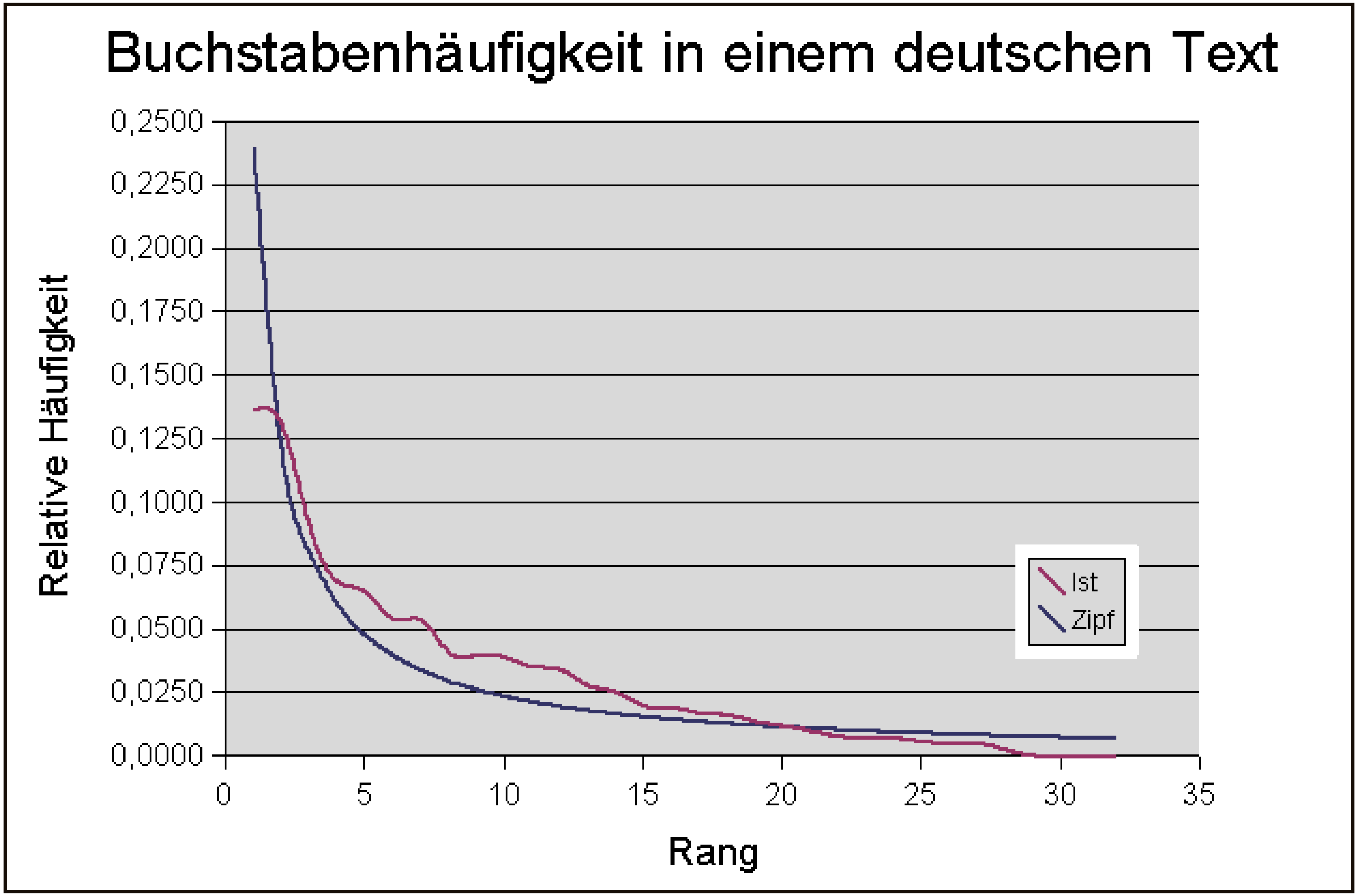
- P – Referenzen (Zugriffe)
- r – Rang (1 = häufigste)
- k – positive Konstante

Mehr dazu: http://www.hpl.hp.com/research/idl/papers/ranking/adamicglottometrics.pdf

# Zipf – Beispiel



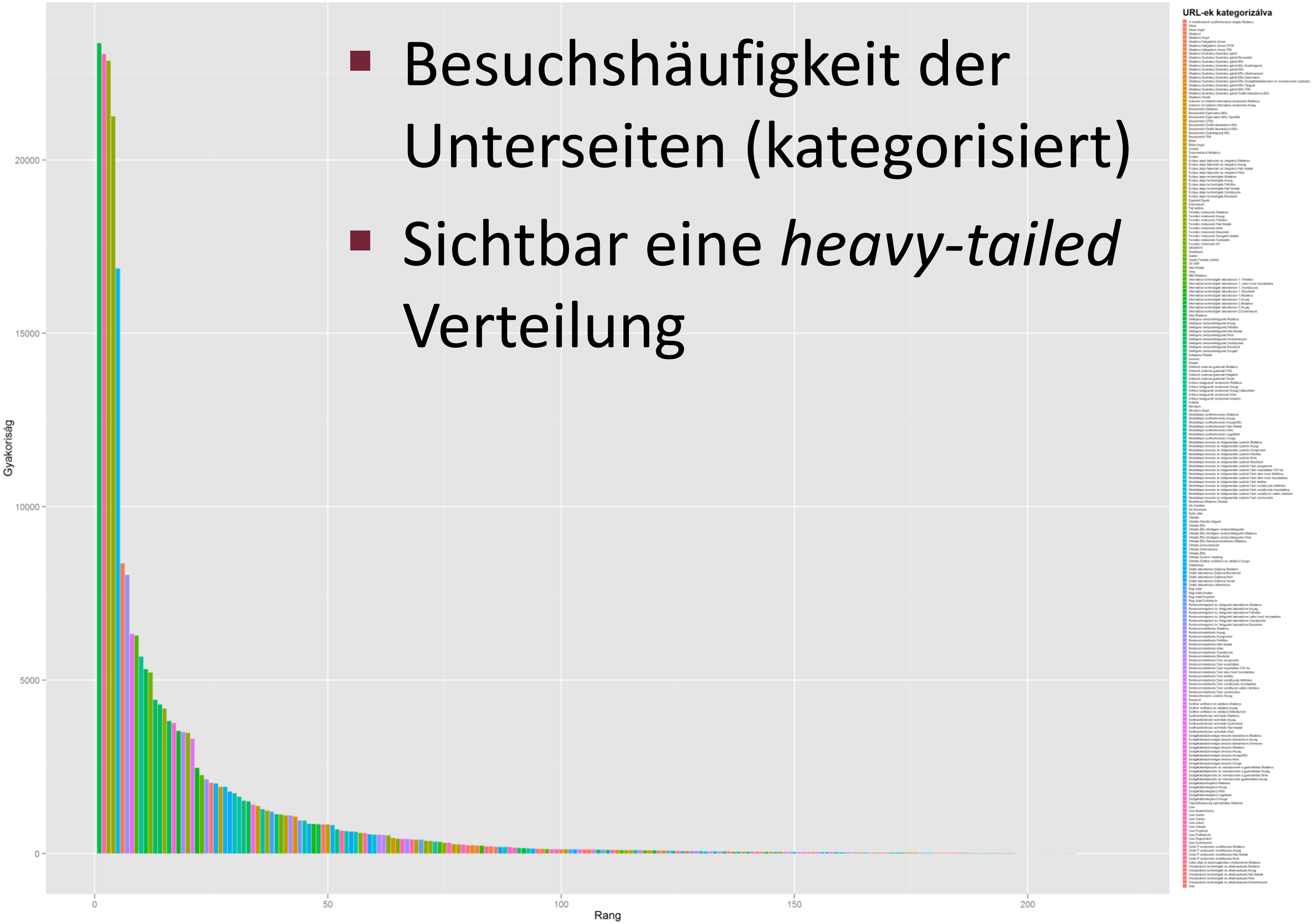
# Zipf – Beispiel




https://upload.wikimedia.org/wikipedia/commons/5/53/Zipf-Verteilung-Buchstaben.png

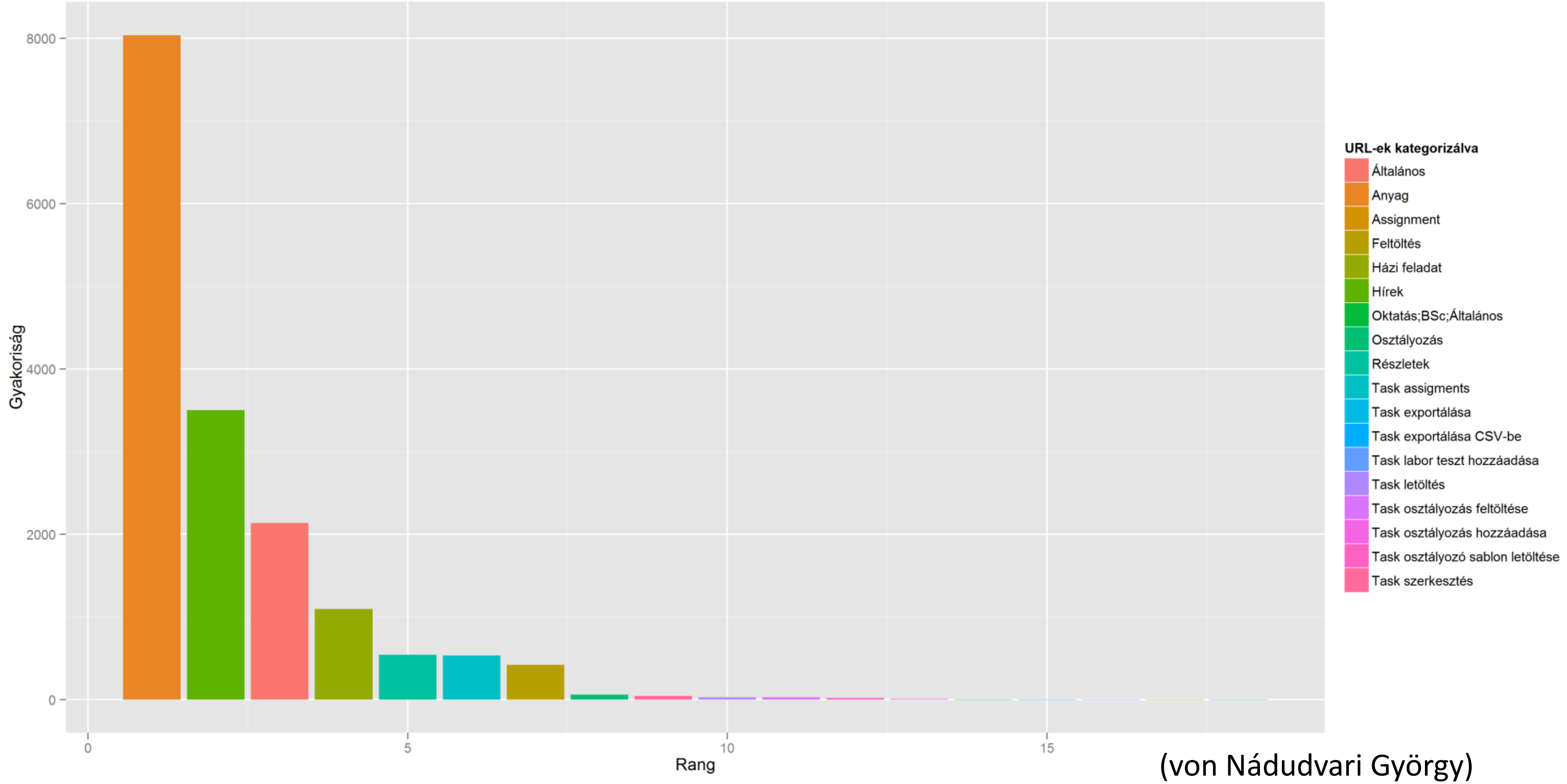
# Zipf – Beispiel: Gruppenwebseite

- Besuchshäufigkeit der Unterseiten (kategorisiert)
- Sichtbar eine *heavy-tailed* Verteilung



# Zipf – Beispiel: Gruppenwebseite

- Besuchshäufigkeit der Seiten der LVA Systemmodellierung




(von Nádudvari György)

# ÄNDERUNGEN DER LAST

# Charakteristiken der Last

- Bisher:
  - Mit Durchschnittswerten gerechnet
  - Das Verhalten des Systems wurde in Abhängigkeit von der *Last(intensität)* betrachtet
  - Die Last nimmt oft nicht (unbedingt) absehbar zu
- In der Wirklichkeit:
  - Das Verhalten des Systems ändert sich *mit der Zeit*
  - Das hat auch technische Folgen
    - Wechseln zwischen den Tasks, Ressourcenreservierung, etc. (z.B. Operationssysteme)

# Änderungen der Last – Beispiel

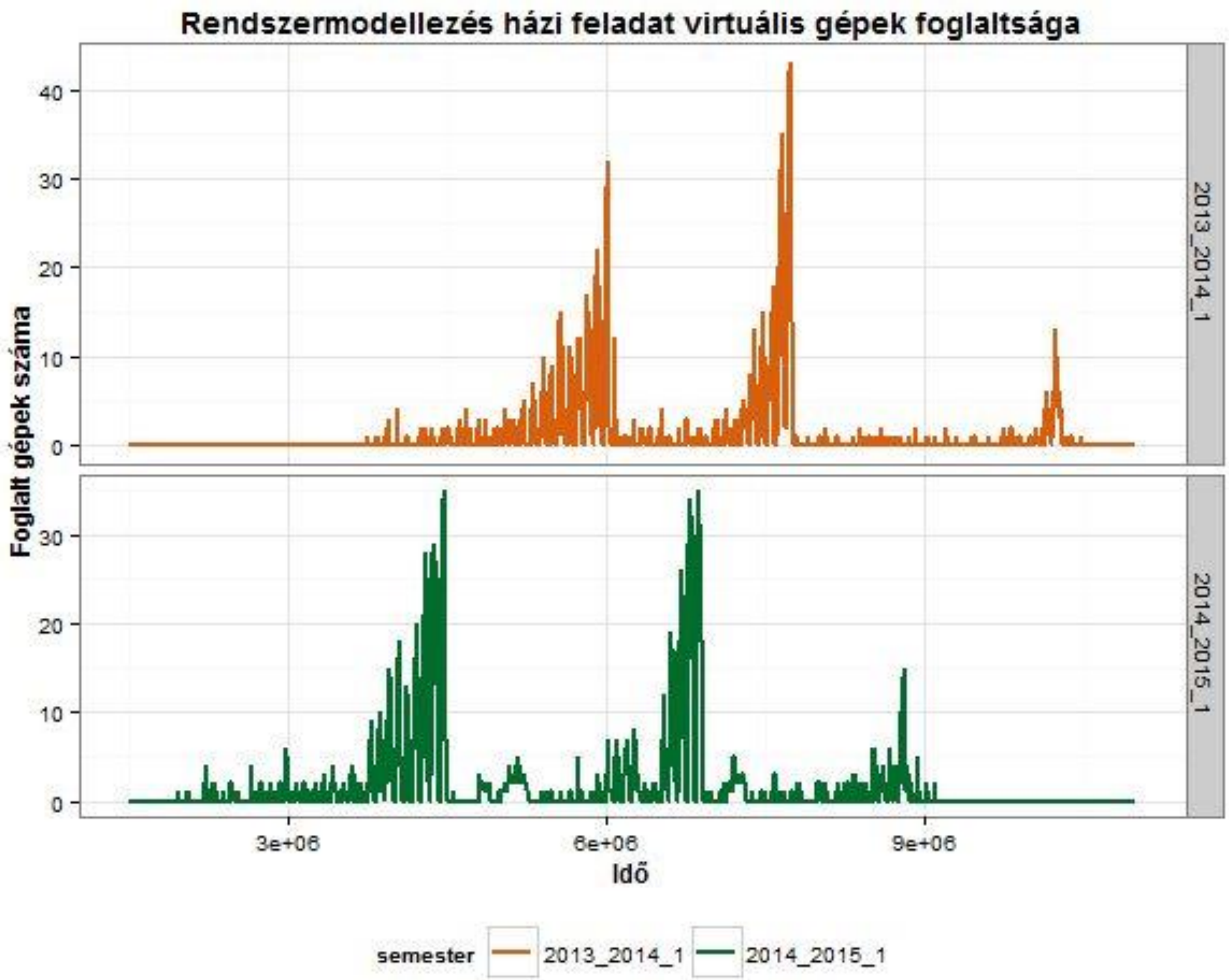
- Dimensionierung des Systems für die Erstellung der (damals) neuen Personalausweise
  - Es ist abschätzbar, wie viele neue Ausweise werden pro Jahr beantragt.
  - Es ist abschätzbar, wie viele Stunden gibt es in einem Jahr.

    → Wir haben einen Wert [Antrag/Stunde]

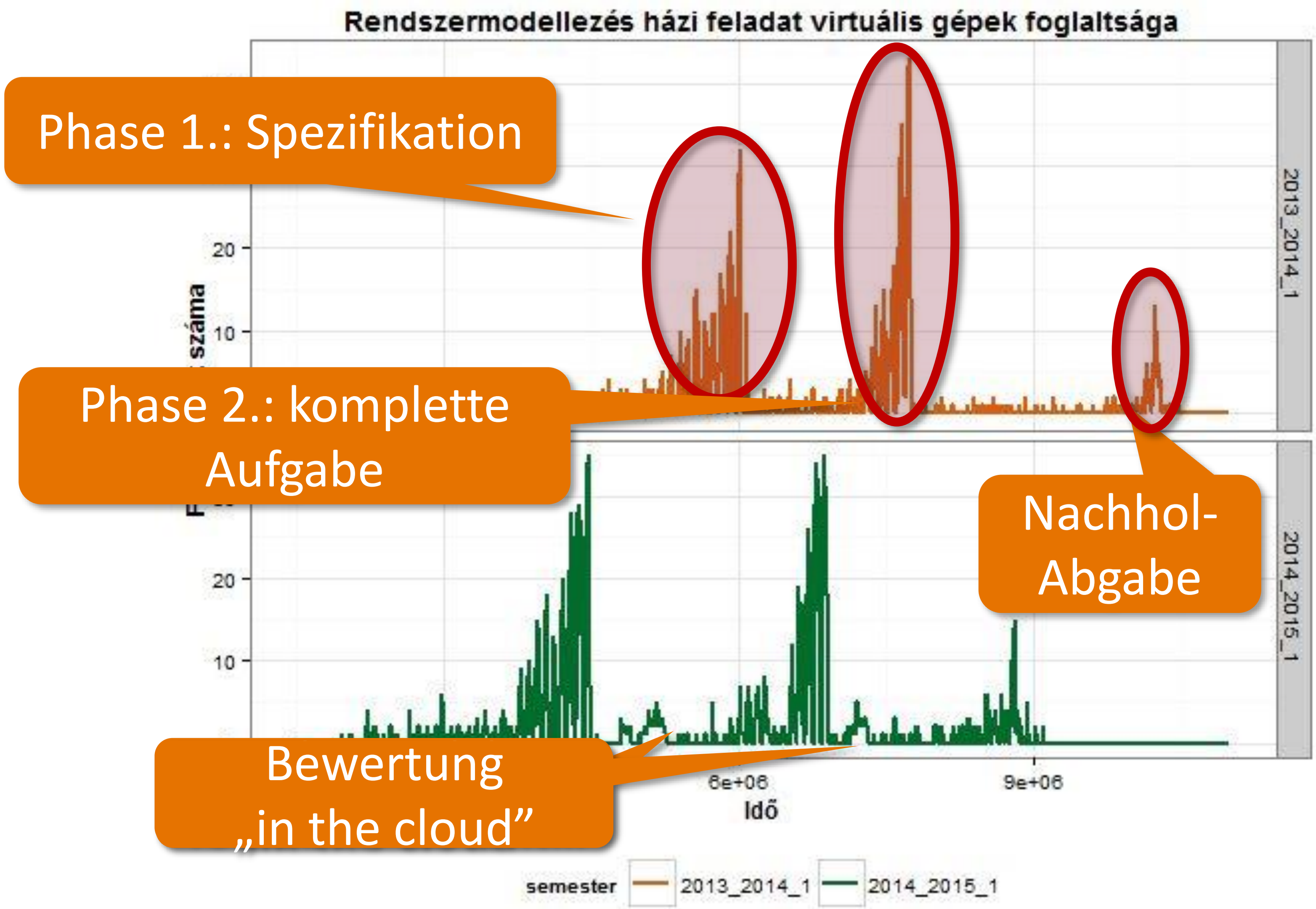
    Kann der als Basis für die Dimensionierung dienen?

- Nehmen wir zwei verschiedene Stunden
  1. 24. Dezember 22-23 Uhr
  2. 15. Juni 16-17 Uhr (Ende des Werktages vor der Haupturlaubszeit)

# Systemmodellierung (7. Semester) – in the cloud



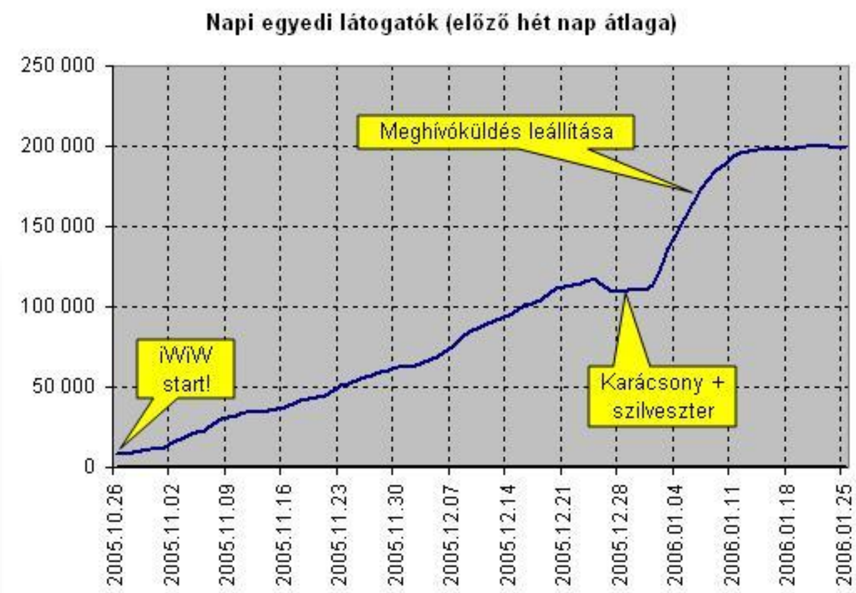
# Systemmodellierung (7. Semester) – in the cloud



# Echte (geschichtliche) Lastdaten (iwiw)


Eine eigene Schätzfunktion für jeden Abschnitt

- Lineare, exponentielle, logarithmisch, …
- Regression, mehr dazu in LVA Wahrscheinlichkeitsrechnung




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