

# Rezept für Welfencreme, erweitert als timed Net

Jan Scholz, Lukas Kozlowski, Fabien Lapok

Hochschule für angewandte Wissenschaften Hamburg,

jan.scholz2@haw-hamburg.de

lukas.kozlowski@haw-hamburg.de

fabien.lapok@haw-hamburg.de

## Aufgabenstellung

Bei der Aufgabe des zweiten Übungstermins handelt es sich um eine Erweiterung der zuvor erhaltenen Aufgabenstellung (Übungstermin eins). Es ist gewünscht das bestehende Netz um ein Timed Net zu erweitern und sich zusätzlich eine geeignete Fragestellung zu überlegen, die mittels einer Simulation beantwortet werden kann.

Unser Netz stellt ein ein Herstellungsprozess eines Rezepts dar. In dem modellierten Prozess werden mehrere Abläufe parallel vollzogen, wodurch uns die Frage sinnvoll erscheint, wie sich die Leistung bei unterschiedlicher Anzahl an Köchen verändert. Diese Fragestellung verfolgt einerseits in dem speziellen Beispiel einen (betriebswirtschaftlichen) Sinn - sowohl die Herstellungszeit, als auch die Personalkosten spielen eine wichtige Rolle in der Gastronomie. Andererseits kann man den Prozess abstrakter als ein nebenläufiges System betrachten, welches von multiplen Prozessoren abgearbeitet wird. Wir erhoffen uns bei der Umsetzung bestimmte Muster zu erkennen, die man aus dem Bereich concurrency in der Informatik kennengelernt hat.

Kommen wir doch zunächst zurück auf die spezielle Fragestellung in der Küche. Um ein optimales Verhältnis an Köchen zu finden, muss aus unserer Sicht zwischen den folgenden beiden Aspekten abgewogen werden: Es ist zum einen schlecht wenn die Zubereitung einer Mahlzeit zu lange dauert, wenn sie frisch für einen Gast hergestellt wird. Und andererseits ist es schlecht wenn Köche aufeinander warten müssen da dann keine effektive Auslastung der Köche gewährleistet ist und weniger Geld erwirtschaftet wird im Verhältnis zu den Personalkosten.

Unter Berücksichtigung beider Aspekte formulieren wir folgende Fragestellung:

“Wie viele Köche müssen beschäftigt werden damit mindestens  
90% aller Gäste, 10min nach ihrer Bestellung, ihre  
Welfencreme bekommen?”

Die Arbeit in einer Küche ist natürlich wesentlich umfangreicher und wird nur zu einem kleinen Teil von unserem Petri-Netz wiedergegeben. Dieser Umstand ist bei der späteren Analyse zu berücksichtigen.

## Methodik

Zunächst müssen die Transitionen, die in dem Petri-Netz aus Übung 1 erstellt wurden genauer untersucht werden. Jede Transition stellt eine Handlung dar die von einem Koch vollzogen werden kann. Zu jeder Handlung muss überlegt werden wie viel Zeit für sie benötigt wird. Renew bietet eine Verteilungsfunktion an mit der sich Zeiten zufällig bereitstellen lassen. Diese Funktion wird zunächst genauer angeschaut (siehe Abbildung 1).

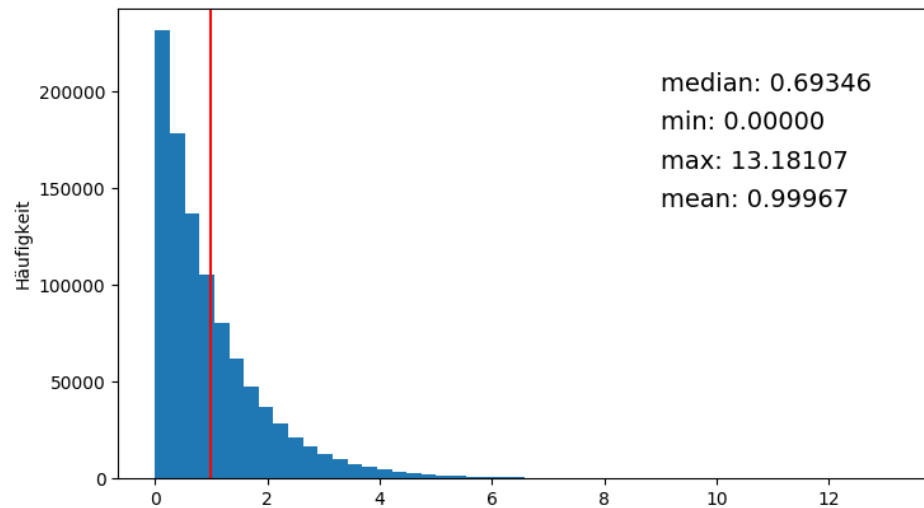


Abbildung 1: Histogramm für die Verteilungsfunktion  $\text{negexp}(1)$  von Renew nach 1 Mio Wiederholungen

Wie der Name der Funktion bereits andeutet handelt es sich um eine negativ exponentiell abfallende Funktion. Aus dem gezeigten Graphen lässt sich ablesen das der Eingabewert der Funktion die durchschnittliche Zeit der Ausgabe darstellt. Die Hälfte aller Ausgabewerte ist unter 0.7 und kompakt auf einer Breite von 0.7 vertreten, die zweite Hälfte von 0.7 bis 13.8 ist wesentlich breiter verteilt mit verschwindend geringer Wahrscheinlichkeit für hohe Werte.

Ähnlich wie bei einem Wurf mehrerer Würfel ergibt sich für die Addition zweier Funktionsaufrufe ein neuer Peak. Der neue Graph (Abbildung 2) ist nun nicht mehr konstant fallend sondern steigt zunächst bis auf sein Maximum und fällt danach ab. Ein ähnliches Verhalten wird in der Simulation ebenfalls, bei der Aneinanderreihung der Handlungen der Köche, erwartet. Diese Eigenschaft scheint intuitiv gut geeignet um die Zeit für diesen Prozess zu berechnen da bei der Herstellung eines Gerichtes einerseits Köche verschieden schnell arbeiten, was zunächst für eine Normalverteilung spricht. Allerdings können auch unvorhergesehene Dinge passieren die den Herstellungsprozess stören wie z.B. das sich ein Koch verletzt, Zutaten nicht vorbereitet sind oder ein Koch gerade eine Zigarette

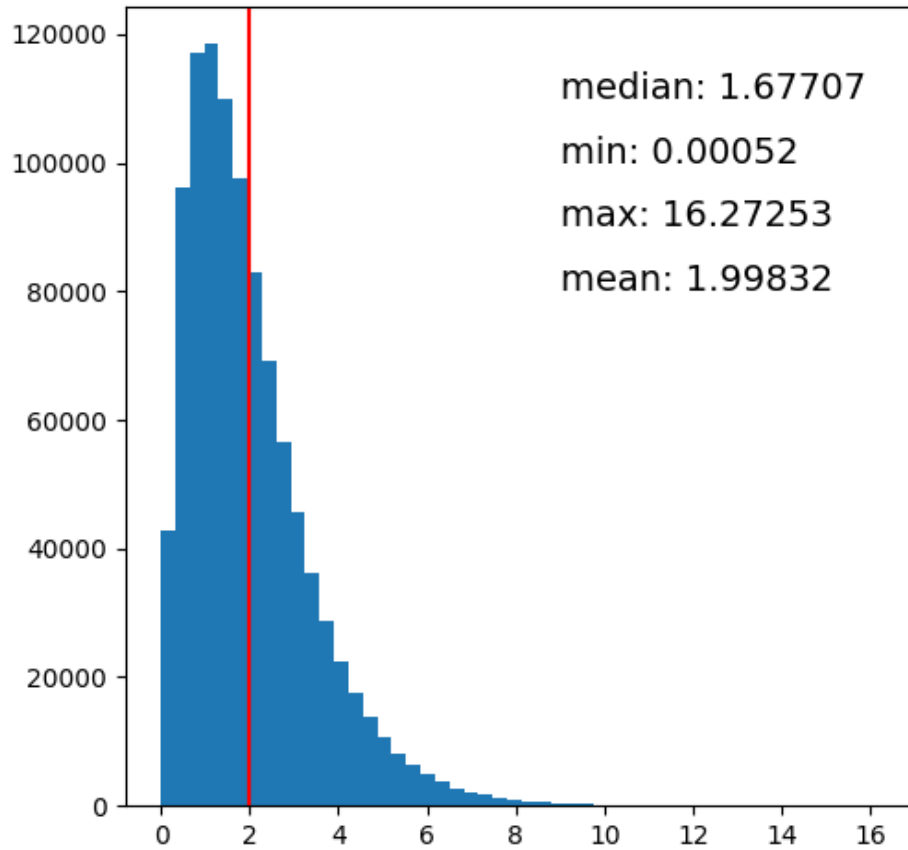


Abbildung 2: Histogramm für  $\text{negexp}(1) + \text{negexp}(1)$  für 1Mio Wiederholungen

rauchen ist. So kann es in Einzelfällen zu wesentlich höheren Herstellungszeiten führen. Mit zunehmender Anzahl von Handlungen wird die Varianz der Zeiten im Verhältnis sinken, denn je mehr Zeiten aufaddiert werden desto geringer ist die Wahrscheinlichkeit das alle Werte sehr hoch oder sehr niedrig werden.

Es muss weiter beachtet werden, dass Handlungen einen minimalen Zeitaufwand haben. Ein Funktionsaufruf von  $\text{negexp}()$  kann auch einen Wert von 0 zurückliefern und die Handlung so direkt nach Antritt abschließen. Dieses Verhalten ist nicht gewünscht und wird verhindert dadurch, dass eine konstante hinzuaddiert wird, die den minimalen Zeitaufwand darstellt. Eine Zeitfunktion sie dann wie folgt aus

$$t_{tr_i} = t_{tr_{i,1}} \cdot \text{Dist.negexp}(t_{tr_{i,2}}) \quad (1)$$

$$t_{tr_i,\mu} = t_{tr_{i,1}} + t_{tr_{i,2}} \quad (\text{Durchschnitt}) \quad (2)$$

$$\tilde{t}_{tr_i} \approx t_{tr_{i,1}} + 0.69 \cdot t_{tr_{i,2}} \quad (\text{Median}) \quad (3)$$

Für die Entwicklung des neuen timed Net's können die Köche als Ressourcen angenommen werden die jeweils nur von einer Transition zur Zeit genutzt werden können.

Wenn für alle Transitionen Zeiten hinterlegt sind kann mit den Simulationen begonnen werden. Die Simulationen werden dann für 1,2,3,4 und 6 Köche jeweils einige male wiederholt bis sich ein repräsentatives Ergebnis ablesen lässt.

## Umsetzung

Tabelle 1 zeigt die von uns approximierten Zeiten für die vorhandenen Transitionen. Für alle Handlungen wird davon ausgegangen das die für die Handlung notwendigen Zutaten und Materialien dem Koch direkt vorliegen und der Koch sofort mit dem Vorgang beginnen kann.

Tabelle 1: Zeiten für Handlungen/Transitionen

Handlung / Transition	Zeit in s
Vanilleschote aufschneiden	$3 + \text{Dist.negexp}(1)$
Mark aufkochen	$180 + \text{Dist.negexp}(5)$
Milch & Stärke verrühren	$5 + \text{Dist.negexp}(4)$
Zitrone & Wein mischen	$5 + \text{Dist.negexp}(1)$
Weinsoße schaumig rühren	$40 + \text{Dist.negexp}(5)$
Milch/Stärke & Vanillemilch verrühren	$5 + \text{Dist.negexp}(1)$
Vanillecreme aufkochen	$60 + \text{Dist.negexp}(10)$
Vanillecreme abkühlen lassen	$260 + \text{Dist.negexp}(40)$
Ei trennen	$5 + \text{Dist.negexp}(1)$
Eiweiss schlagen	$20 + \text{Dist.negexp}(4)$
Schnee unter Vanillecreme unterheben	$5 + \text{Dist.negexp}(5)$
Vanillecreme + Schnee abfüllen	$2 + \text{Dist.negexp}(2)$
Vanillecreme mit heißer Weinsoße auffüllen	$2 + \text{Dist.negexp}(2)$

Das erweiterte Petrinetz sieht wie in Abbildung 3 dargestellt aus. Es wurde ein *Hilfs-Platz* namens Tasse hinzugefügt, die dazu dient, dass die Stärke nicht doppelt für die Herstellung von dem Milch-Stärke-Gemisch verwendet wird, und das System verklemmt. Zusätzlich wurde ein Platz namens *Koch* hinzugefügt, der eine Kante zu allen Transitionen die eine Tätigkeit eines Kochs repräsentieren hat. An diesem Platz wird die Anzahl der Köche durch die Startmarkierung verändert.

Die Durchführung haben wir mithilfe eines von uns geschriebenen Autohotkey-Skripts automatisiert, worüber das Starten und Stoppen getriggert wurde. Die Datenausgabe haben wir über den Strat von Renew über den folgenden Aufruf in eine Textdatei schreiben lassen:

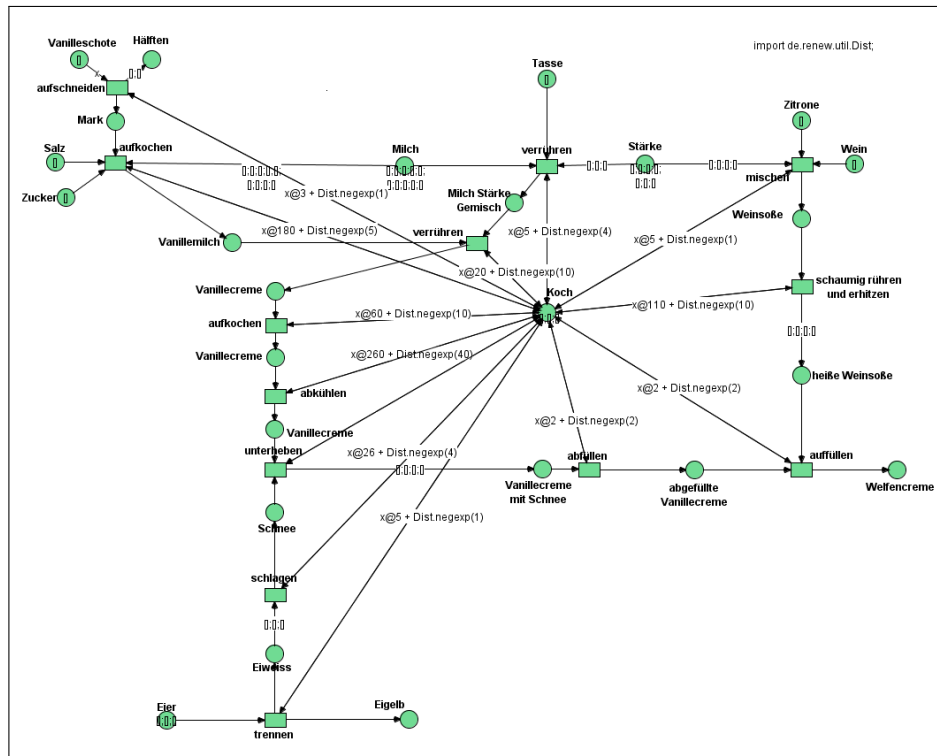


Abbildung 3: Das auf Basis der Rezeptur erstellte Petrinetz, erweitert zum timed Net

```
java -jar loader.jar gui > data.txt
```

## Analyse

Das Resultat ist in Abbildung 4 dargestellt. Die dunkel gestrichelte Linie ist repräsentiert die neunte Dezille bezogen auf die durch die Farbe repräsentierte Anzahl an Köchen. Die zuvor definierte Frage *Wie viele Koche müssen beschäftigt werden damit mindestens 90% aller Gäste, 10min nach ihrer Bestellung, ihre Welfencreme bekommen?* lässt sich mit *es werden drei Köche benötigt* beantworten.

*Beobachtungen* Aus betriebswirtschaftlicher Sicht ist es evtl. besser sich nur für zwei Köche zu entscheiden, weil der zeitliche Nachteil gegenüber drei Köche nicht groß ist (ungefähr 20 sec. bei der neunten Dezille). Eine weitere interessante Beobachtung ist, dass die größte zeitliche Verbesserung bei dem Sprung von einem auf zwei Köche zu verzeichnen ist. Der Median sinkt von 866,9 auf 575,5 Sekunden. Betrachtet man dazu im Vergleich die Beschleunigung beim Einsatz von

sechs, statt vier Köche, hat man nur eine Verbesserung von ca. zwölf Sekunden im Median. Dieses Verhalten kann man auch in der IT, beim Einsatz von mehreren Prozessoren beobachten. Sofern die Anwendung nicht auf mehr Prozessoren ausgelegt ist, ist die Performanceverbesserung durch den Einsatz von mehr Prozessoren nur beschränkt.

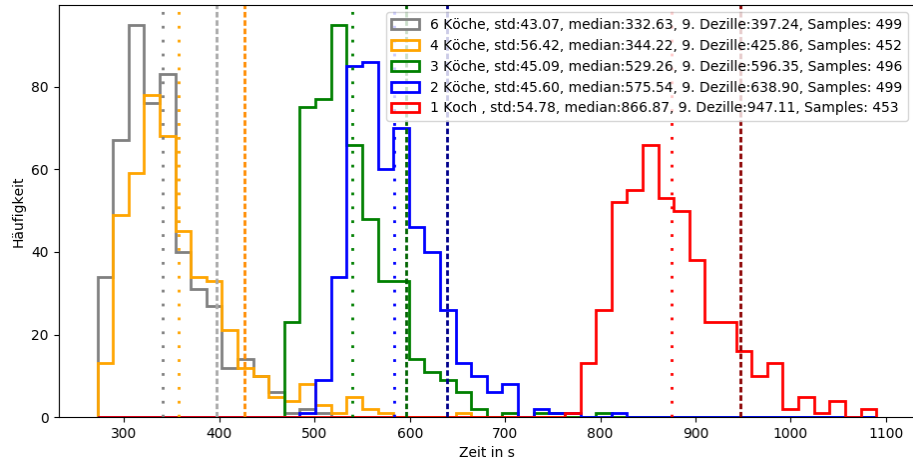


Abbildung 4: Histogramm: Die Simulation des Petrinetzes mit verschiedener Anzahl Köchen, gepunktet der korrespondierende Median, dunkel gestrichelt 9. Dezille