

# **Präsentation Bachelor Arbeit: Simulation von Online Scheduling-Algorithmen für parallele Systeme**

---

Hendrik Rassmann

November 02, 2020

TU Dortmund University - Department of Computer Science

Problemstellung

Simulationsaufbau

Outlook

## A comparative study of online scheduling algorithms for networks of workstations

Olaf Arndt<sup>a</sup>, Bernd Freisleben<sup>a</sup>, Thilo Kielmann<sup>b</sup> and Frank Thilo<sup>a</sup>

<sup>a</sup> *Department of Electrical Engineering and Computer Science, University of Siegen, Germany*

<sup>b</sup> *Department of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, The Netherlands*

Networks of workstations offer large amounts of unused processing time. Resource management systems are able to exploit this computing capacity by assigning compute-intensive tasks to idle workstations. To avoid interferences between multiple, concurrently running applications, such resource management systems have to schedule application jobs carefully. Continuously arriving jobs and dynamically changing amounts of available CPU capacity make traditional scheduling algorithms difficult to apply in workstation networks. *Online scheduling algorithms* promise better results by adapting schedules to changing situations. This paper compares six online scheduling algorithms by simulating several workload scenarios. Based on the insights gained by simulation, the three online scheduling algorithms performing best were implemented in the WINNER resource management system. Experiments conducted with WINNER in a real workstation network confirm the simulation results obtained.

- Network
- Online
- Experiments ... confirm

# Problemstellung

---

Ausgehend von:

- Rechner/*Nodes* unterschiedlicher *Geschwindigkeiten* bilden zusammen ein (Rechen)*Cluster*

Ausgehend von:

- Rechner/*Nodes* unterschiedlicher *Geschwindigkeiten* bilden zusammen ein (Rechen)*Cluster*
- Aufträge/*Jobs* kommen im laufenden Betrieb rein (*online*)

Ausgehend von:

- Rechner/*Nodes* unterschiedlicher *Geschwindigkeiten* bilden zusammen ein (Rechen)*Cluster*
- Aufträge/*Jobs* kommen im laufenden Betrieb rein (*online*)
- Aufträge unterscheiden sich bez. *Bearbeitungszeit*, *Anzahl* an benötigten Knoten und *Ankunftszeitpunkt*

Ausgehend von:

- Rechner/*Nodes* unterschiedlicher *Geschwindigkeiten* bilden zusammen ein (Rechen)*Cluster*
- Aufträge/*Jobs* kommen im laufenden Betrieb rein (*online*)
- Aufträge unterscheiden sich bez. *Bearbeitungszeit*, *Anzahl* an benötigten Knoten und *Ankunftszeitpunkt*
- Aufträge sollen auf eine 'geeignete' Art und Weise bearbeitet werden



- *makespan* "Zeit bis Feierabend"

---

<sup>1</sup>Kriterien von Arndt. et al.

- *makespan* "Zeit bis Feierabend"
- *average flow time* "Supermarkt - Nur der Apfel? Gehen Sie doch gerne vor"

---

<sup>1</sup>Kriterien von Arndt. et al.

- *makespan* "Zeit bis Feierabend"
- *average flow time* "Supermarkt - Nur der Apfel? Gehen Sie doch gerne vor"
- *average waiting time*

---

<sup>1</sup>Kriterien von Arndt. et al.

- *makespan* "Zeit bis Feierabend"
- *average flow time* "Supermarkt - Nur der Apfel? Gehen Sie doch gerne vor"
- *average waiting time*
- *maximum waiting time* "Restaurant - Wenn ein Tisch nicht bedient wird, kommen Gäste nicht mehr wieder"

---

<sup>1</sup>Kriterien von Arndt. et al.

*Scheduling-Algorithmus : Warteliste  $\rightarrow$  Auftrag*

---

<sup>2</sup>Auswahl von Arndt. et al.

*Scheduling-Algorithmus : Warteliste  $\rightarrow$  Auftrag*

- First in First out (*FiFo*)

---

<sup>2</sup>Auswahl von Arndt. et al.

*Scheduling-Algorithmus : Warteliste  $\rightarrow$  Auftrag*

- First in First out (*FiFo*)
- Shortest Processing Time first (*SPT*)

---

<sup>2</sup>Auswahl von Arndt. et al.

*Scheduling-Algorithmus : Warteliste  $\rightarrow$  Auftrag*

- First in First out (*FiFo*)
- Shortest Processing Time first (*SPT*)
- Greatest Processing Time first (*GPT*)

---

<sup>2</sup>Auswahl von Arndt. et al.



Q:

Dauer =2, Knoten =1

Dauer =4, Knoten =1

Dauer =3, Knoten =2

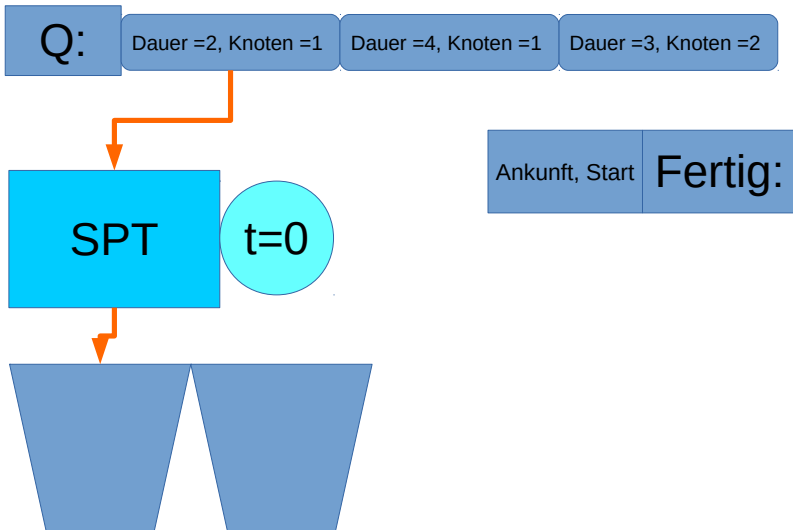
SPT

t=0

Ankunft, Start

Fertig:





Q:

Dauer =4, Knoten =1

Dauer =3, Knoten =2

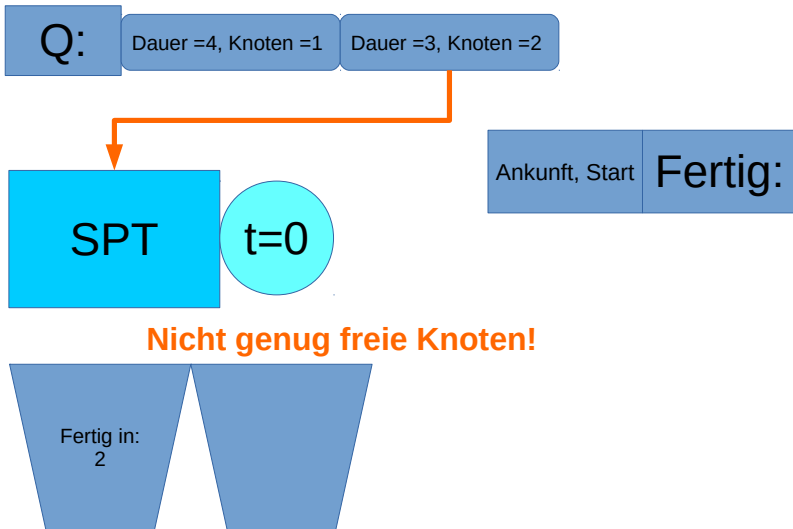
SPT

t=0

Ankunft, Start

Fertig:

Fertig in:  
2



Q:

Dauer =4, Knoten =1

Dauer =3, Knoten =2

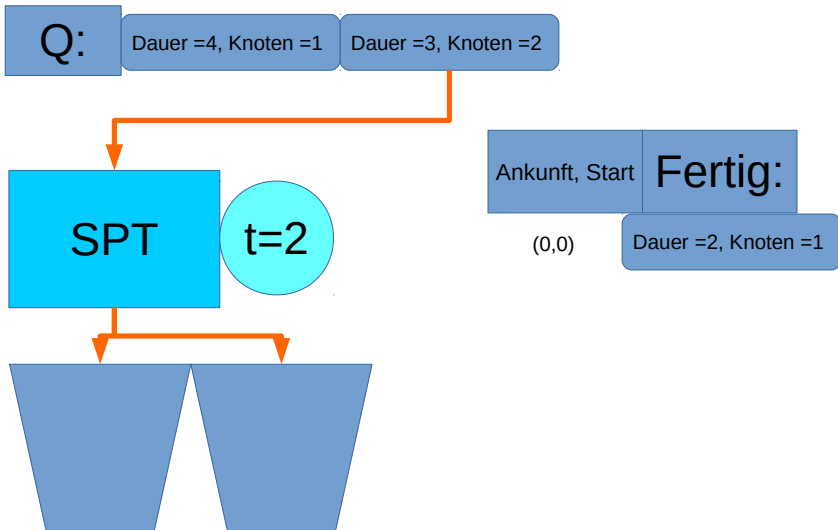
SPT

t=1

Ankunft, Start

Fertig:

Fertig in:  
1



Q:

Dauer =4, Knoten =1

SPT

t=2

Fertig in:  
2

Fertig in:  
2

Ankunft, Start

Fertig:

(0,0)

Dauer =2, Knoten =1

Q:

Dauer =4, Knoten =1

SPT

t=3

Fertig in:  
1

Fertig in:  
1

Ankunft, Start

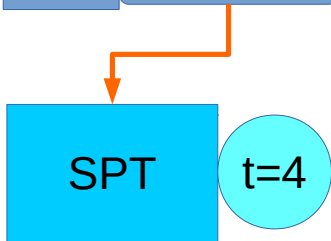
Fertig:

(0,0)

Dauer =2, Knoten =1



Q: Dauer =4, Knoten =1



Ankunft, Start	Fertig:
(0,0)	Dauer =2, Knoten =1
(0,2)	Dauer =3, Knoten =2

Q:

SPT

t=4

Fertig in:  
4

Ankunft, Start

Fertig:

(0,0)

Dauer =2, Knoten =1

(0,2)

Dauer =3, Knoten =2

Q:

SPT

t=5

Fertig in:  
3

Ankunft, Start

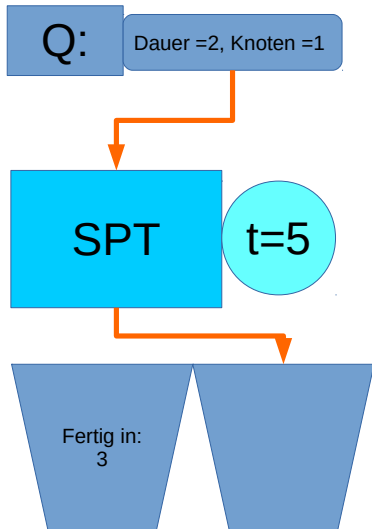
Fertig:

(0,0)

Dauer =2, Knoten =1

(0,2)

Dauer =3, Knoten =2



Ankunft, Start	Fertig:
(0,0)	Dauer =2, Knoten =1
(0,2)	Dauer =3, Knoten =2

Q:

SPT

t=5

Fertig in:  
3

Fertig in:  
2

Ankunft, Start

Fertig:

(0,0)

Dauer =2, Knoten =1

(0,2)

Dauer =3, Knoten =2

Q:

SPT

t=6

Fertig in:  
2

Fertig in:  
1

Ankunft, Start

Fertig:

(0,0)

Dauer =2, Knoten =1

(0,2)

Dauer =3, Knoten =2

Q:

SPT

t=7

Fertig in:  
1

Ankunft, Start

Fertig:

(0,0)

Dauer =2, Knoten =1

(0,2)

Dauer =3, Knoten =2

(5,5)

Dauer =2, Knoten =1

Q:

SPT

t=8



Ankunft, Start

Fertig:

(0,0)

Dauer =2, Knoten =1

(0,2)

Dauer =3, Knoten =2

(5,5)

Dauer =2, Knoten =1

(0,4)

Dauer =4, Knoten =1



## Scheduling-Algorithmen (Referentiell Intransparent)

- Random

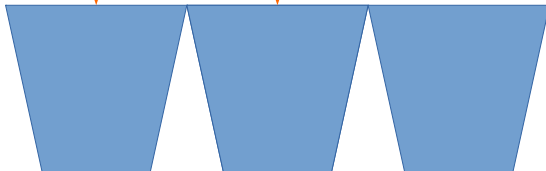
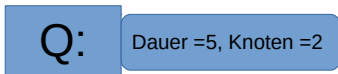
## Scheduling-Algorithmen (Referentiell Intransparent)

- Random
- **First Fit** (Wähle den am längsten wartende Auftrag, der sofort gestartet werden kann)

## Scheduling-Algorithmen (Referentiell Intransparent)

- Random
- **First Fit** (Wähle den am längsten wartende Auftrag, der sofort gestartet werden kann)
- **Backfilling** (Wähle einen startbaren Auftrag, der wenn er jetzt gestartet wird, terminiert bevor der am längsten wartende Auftrag starten wird)

## Beispiel Backfilling



Q:

Backfillig

t=0

Ankunft, Start

Fertig:

Fertig in:  
5

Fertig in:  
5

Q:

Dauer =3, Knoten =2

Backfillig

t=1

Ankunft, Start

Fertig:

Fertig in:  
4

Fertig in:  
4

Q:

Dauer =3, Knoten =2

Dauer =1, Knoten =1

Backfillig

t=2

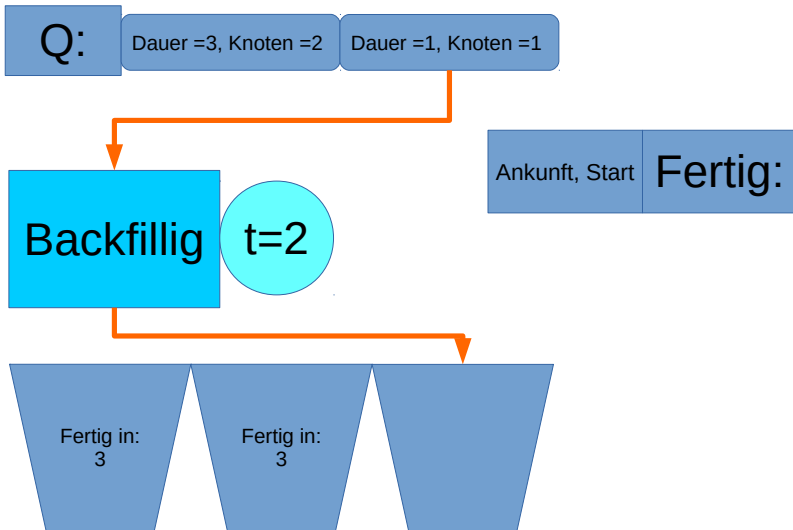
Ankunft, Start

Fertig:

Fertig in:  
3

Fertig in:  
3





Q:

Dauer =3, Knoten =2

Backfillig

t=2

Ankunft, Start

Fertig:

Fertig in:  
3

Fertig in:  
3

Fertig in:  
1

Q:

Dauer =3, Knoten =2

Backfillig

t=3

Ankunft, Start

Fertig:

Fertig in:  
2

Fertig in:  
2

Q:

Dauer =3, Knoten =2

Dauer =8, Knoten =1

Backfillig

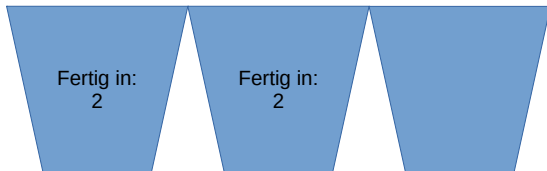
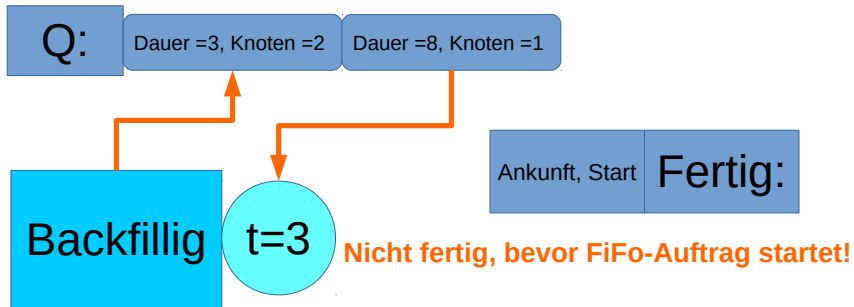
t=3

Ankunft, Start

Fertig:

Fertig in:  
2

Fertig in:  
2



# Simulationsaufbau

---

- Aktivitätsorientiert (Wettersystem, Physik Simulation in Videospielen)
- Ereignisorientiert (Fahrplan)
- Prozessorientiert (Parallele Programmierung)

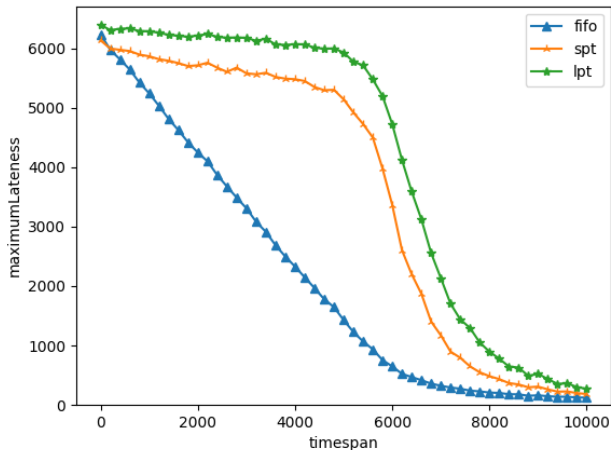
---

<sup>3</sup>Matloff, Norm: Introduction to discrete-event simulation and the simpy language. In: Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August 2 (2008), Nr. 2009, S. 133

- Arndt et al. **designen 10 Experimente**, um unterschiedliches Verhalten der Algorithmen zu demonstrieren
- Variieren einen Parameter und vergleichen gewählte Zielfunktionen



# Beispiel



- 250 Aufträge, 1-50 k s Bearbeitungszeit
- 10 Knoten
- timespan - Spätester Ankunftszeitpunkt wird variiert
- Mittel von 100 Läufen

## Outlook

---

The font size should be as large as possible

- *Never smaller than the age of the oldest audience member*
- Don't vary sizes too much, never within a slide
- Stick to one font
- Don't overuse formatting like bold, italics, alert, etc.

But you knew all that already...

## Querying Data used to be simple...

- Tutorial for Beamer [https://www.overleaf.com/learn/latex/Beamer\\_Presentations:\\_A\\_Tutorial\\_for\\_Beginners\\_\(Part\\_1\)%E2%80%94Getting\\_Started](https://www.overleaf.com/learn/latex/Beamer_Presentations:_A_Tutorial_for_Beginners_(Part_1)%E2%80%94Getting_Started)
- Quelle Template: <https://github.com/matze/mtheme>
- Quelle Bild: <https://unsplash.com/photos/52gEprMkp7M>

## How about a very large image?



- Itemize Environments are important
- We can highlight words

## How about a very large image?



- Itemize Environments are important
- We can highlight words
- We can add animations.  
(please don't over-do it)

## **Clever Last Words to Stimulate Discussion**