

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^a, Bernd Freisleben^a, Thilo Kielmann^b and Frank Thilo^a

^a *Department of Electrical Engineering and Computer Science, University of Siegen, Germany*

^b *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

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"

¹Kriterien von Arndt. et al.

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

¹Kriterien von Arndt. et al.

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

¹Kriterien 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"

¹Kriterien von Arndt. et al.

Scheduling-Algorithmen (Referentiell Transparent)²

Scheduling-Algorithmus : Warteliste \rightarrow Auftrag

²Auswahl von Arndt. et al.

Scheduling-Algorithmen (Referentiell Transparent)²

Scheduling-Algorithmus : Warteliste \rightarrow Auftrag

- First in First out (*FiFo*)

²Auswahl von Arndt. et al.

Scheduling-Algorithmen (Referentiell Transparent)²

Scheduling-Algorithmus : Warteliste \rightarrow Auftrag

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

²Auswahl von Arndt. et al.

Scheduling-Algorithmen (Referentiell Transparent)²

Scheduling-Algorithmus : Warteliste \rightarrow Auftrag

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

²Auswahl von Arndt. et al.

Scheduling-Algorithmen (Referentiell Intransparent)

content...

Simulationsaufbau

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`

- 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