

# Praktikum: Parallele Programmierung

## **Parallele Ballerkennung (veraltet!)**

Michael Straßberger; Philip Gawehn

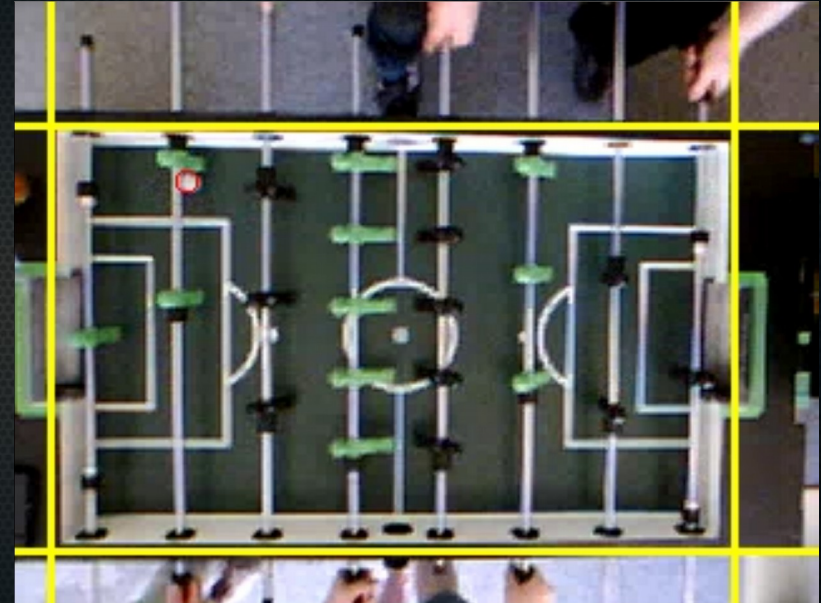
# Ziel (Recap)

- 1. Spielfeld erkennen
  - relativ einfach möglich
- 2. Ball erkennen
  - schwieriger



# Warum?

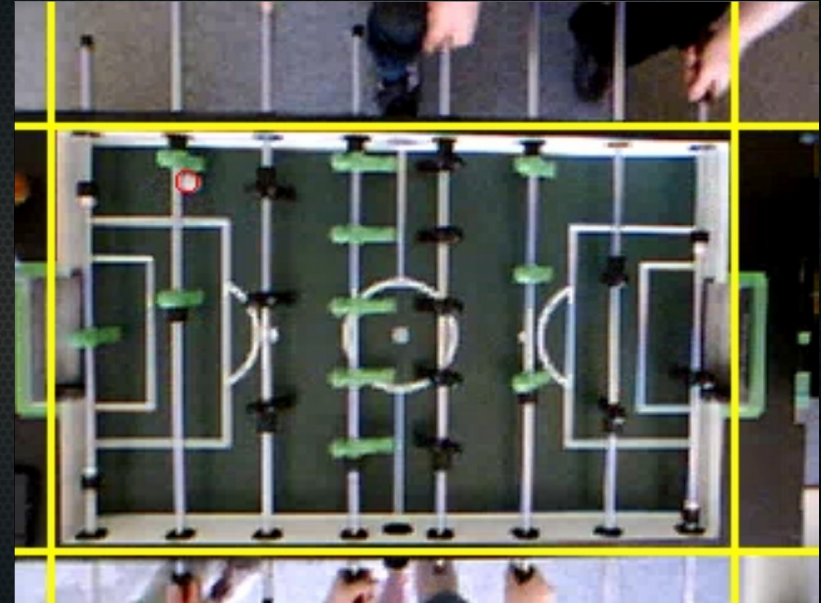
- 1. Spielfeld erkennen
  - Suche nach langen, schwarzen Kanten.
  - Auf Basis dieser werden die anderen beiden Kanten berechnet





# Warum?

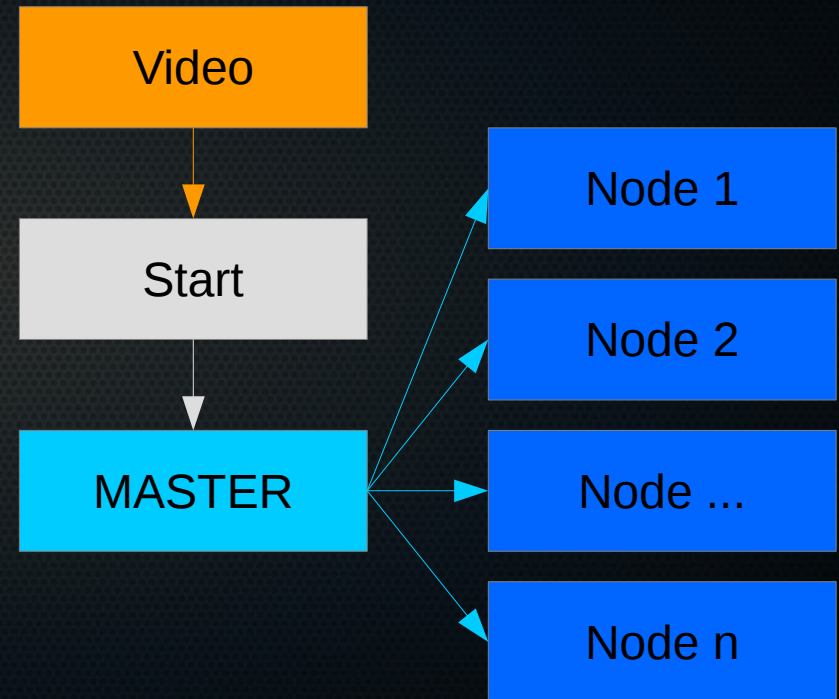
- 2. Ball erkennen
  - Suche innerhalb des erkannten Spielfeldes
  - Muss noch optimiert werden



So weit, so einfach...  
Jetzt wird's **parallel!**

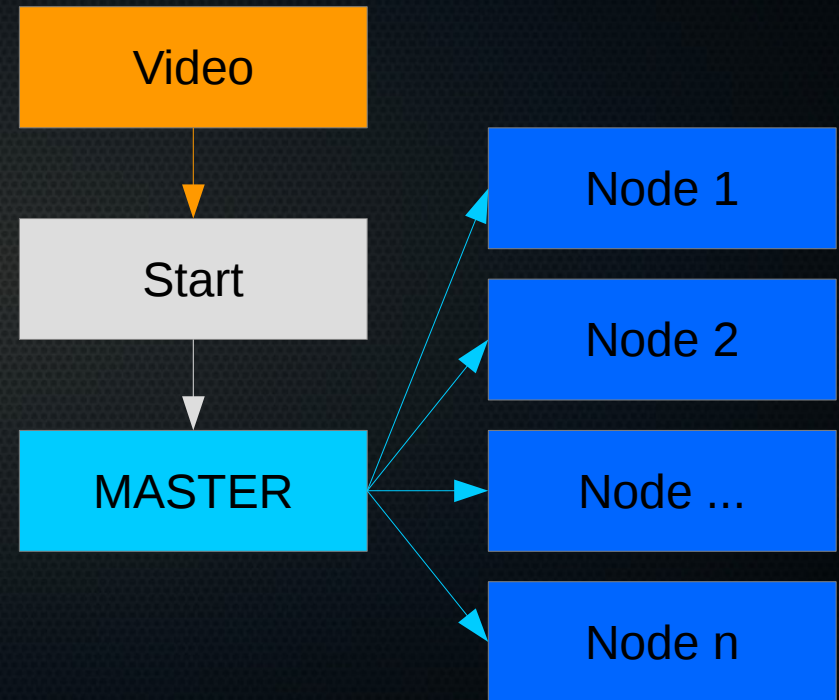
# Parallel: I

- Das Video wird allen Nodes durch Shared Memory zur Verfügung gestellt.
- MASTER koordiniert die Aufgabenverteilung



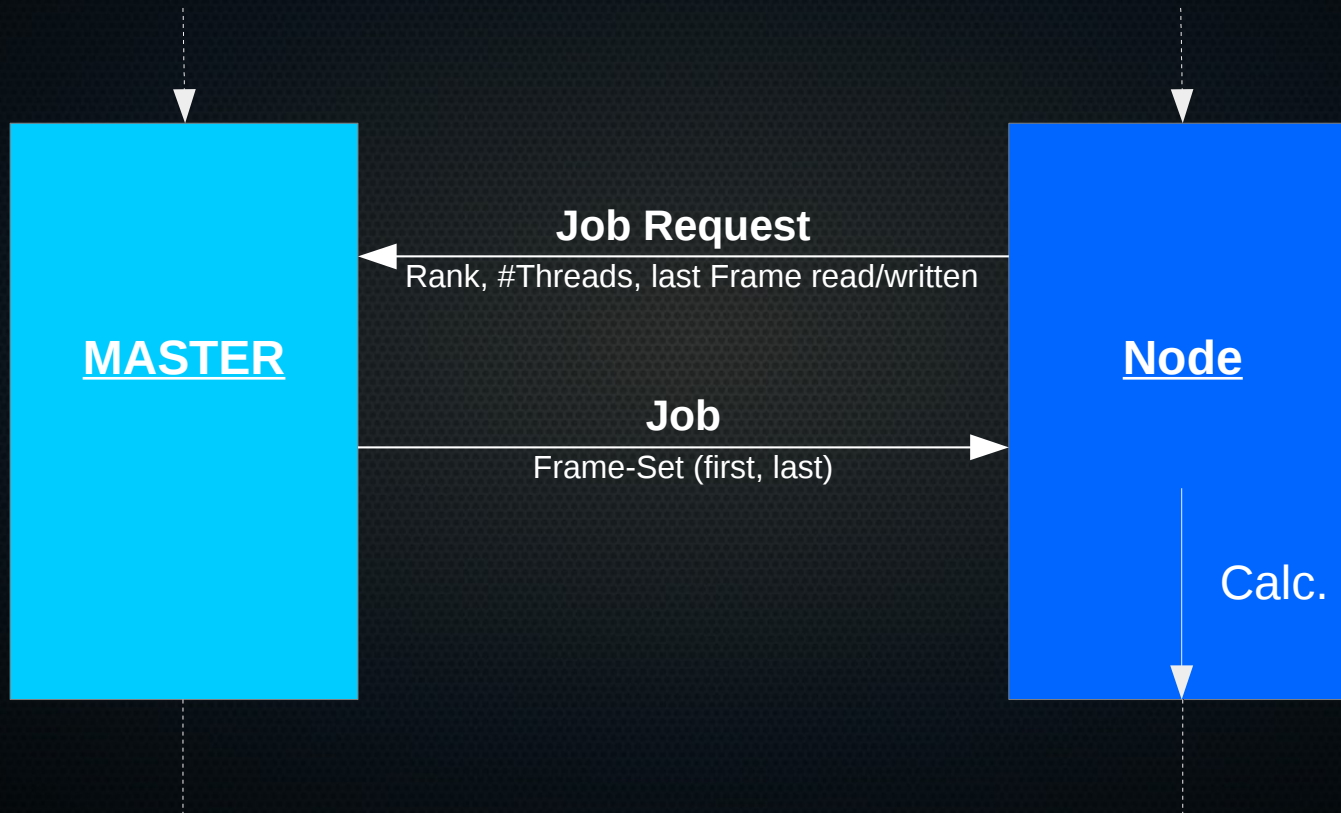
# Parallel: II

- Einzelne Nodes beantragen beim MASTER neue Packets (Frame-Sets)
- MASTER teilt die Packets je nach Leistung (Threads) der Nodes ein.





# Parallel: III

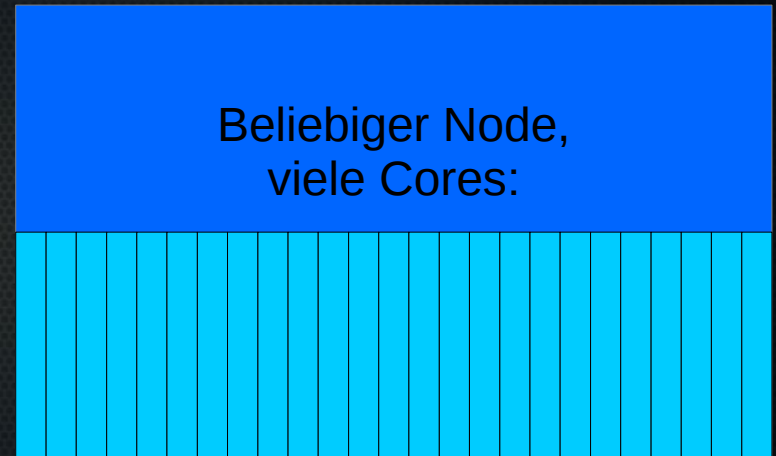


\*vereinfachte Darstellung\*



# Parallel: IV

- Die einzelnen Nodes nutzen OpenMP, um die Berechnung zu parallelisieren.
- Einzelne Frame-Packets werden am Ende zusammengeschnitten und vom MASTER wieder auf die Festplatte geschrieben.



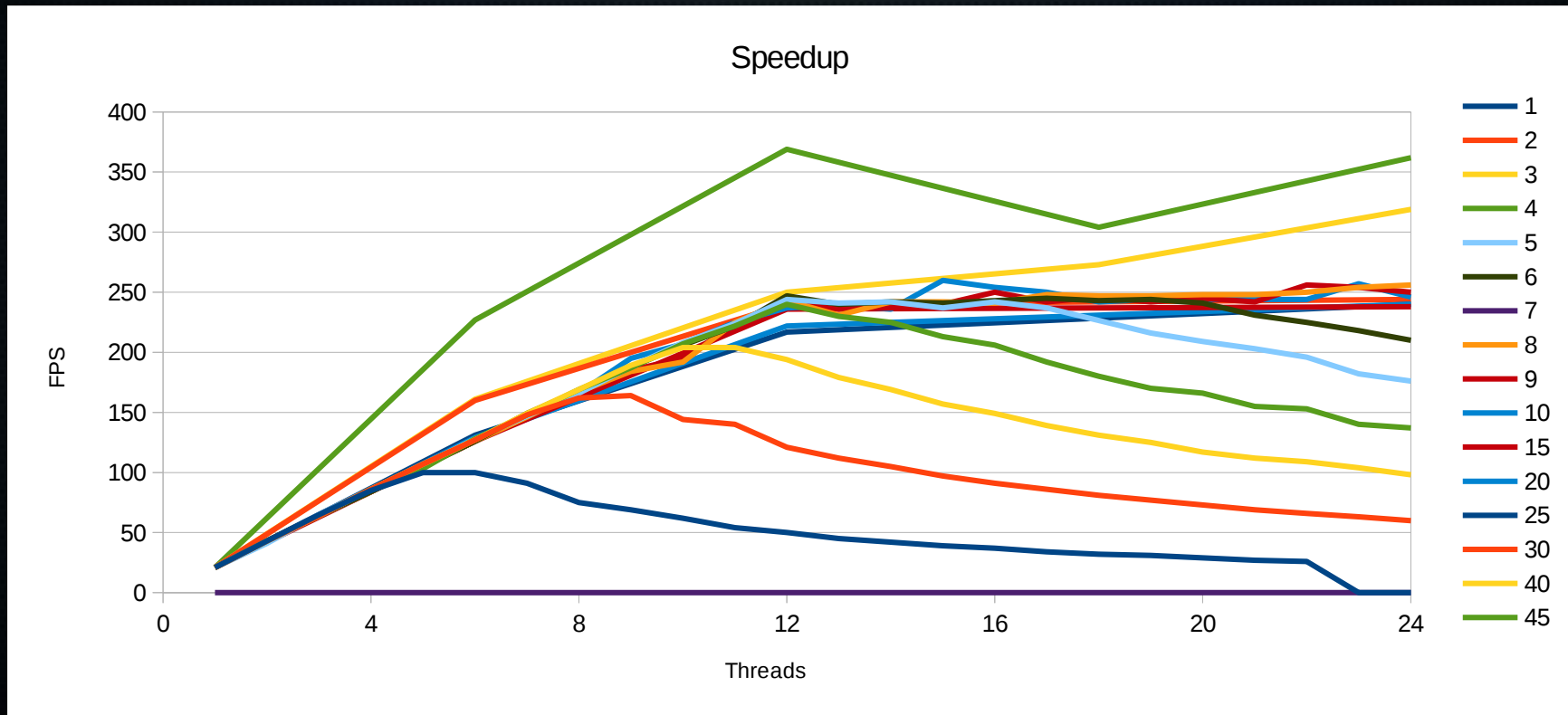
# Parallel: V

- Das Problem...
  - Einzelne Nodes betrachten Packets (Frame-Sets)
  - Können nicht auf Ergebnisse und Statistiken der anderen Packets zugreifen
  - Konsequenz
    - Packets sind abgeschlossen
    - Wir betrachten jeden Frame einzeln
    - Macht die Erkennung komplizierter

Speedup ;)



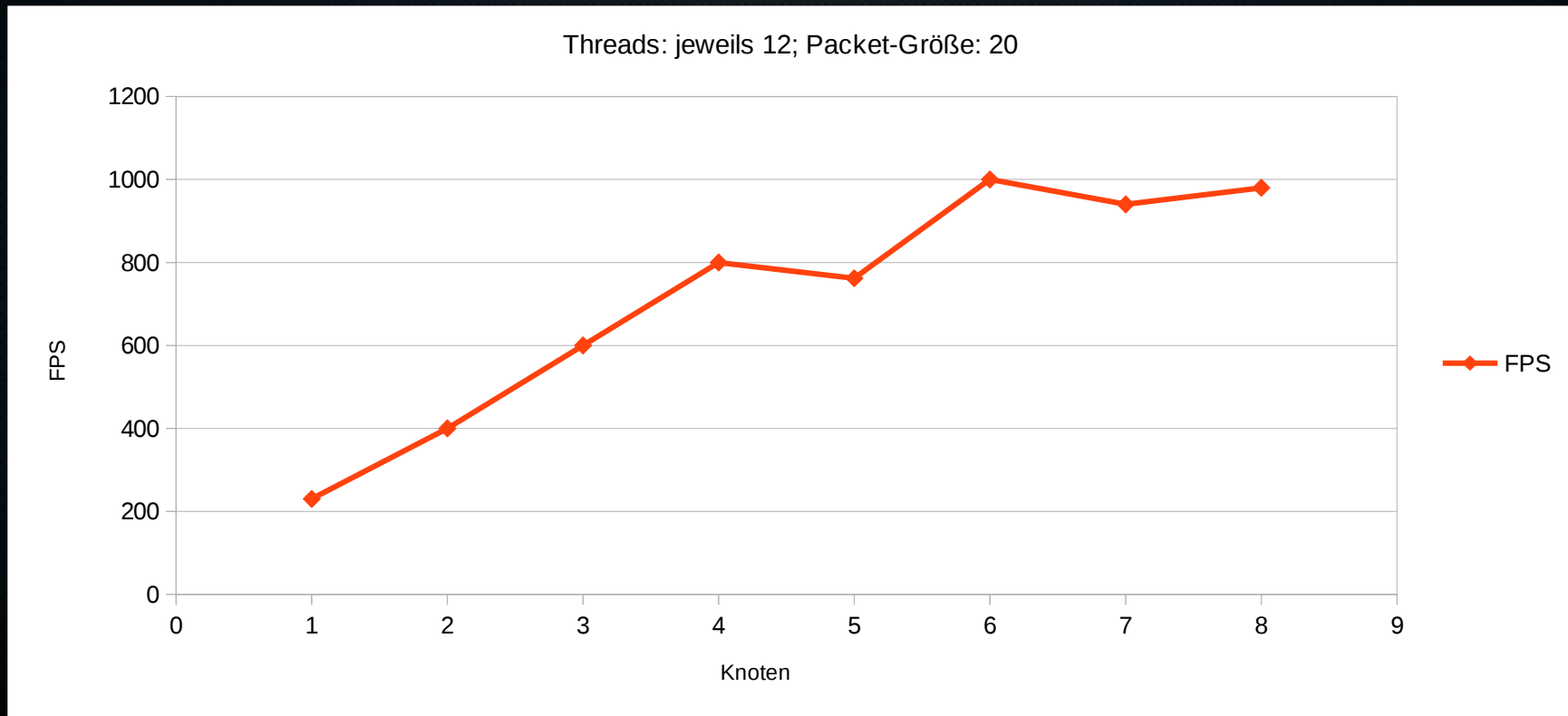
# Speedup I



# Speedup II

Platz	Threads	Packet Size	FPS
#1	12	45	369
#2	24	45	362
#3	24	40	319
#4	18	45	304
#5	18	40	273

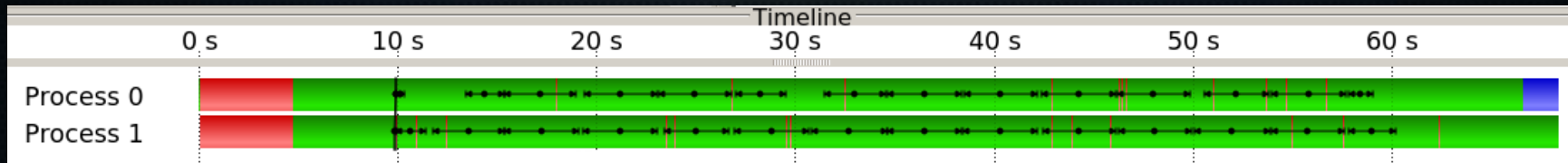
# Speedup III





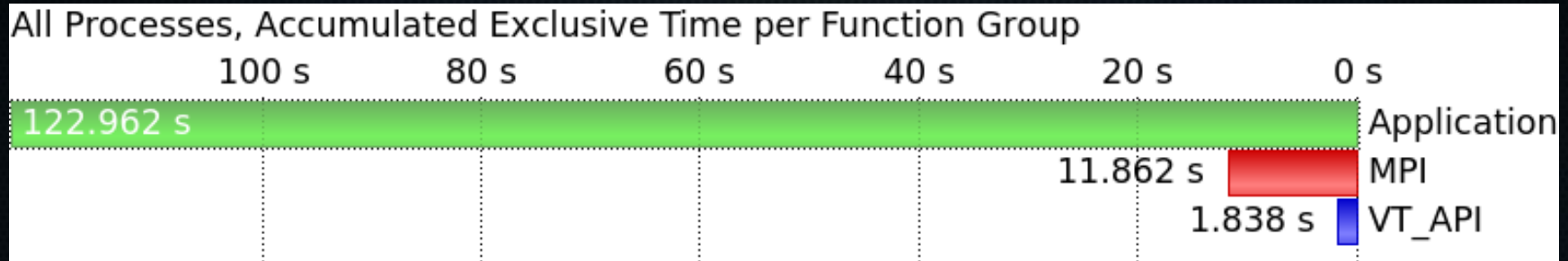
Vampir

# Vampir I



- Sehr wenig MPI Kommunikation nach Initialisierung
- Sehr gute Auslastung der Knoten

# Vampir II



- Nur 9,64688277679283 % MPI
- Etwa 6 % ohne MPI\_Init
- sehr gut :)



That's it!

**Habt ihr noch Fragen?**

Michael Straßberger; Philip Gawehn