



<http://kinderuni.fh-landshut.de/wp-content/uploads/2011/01/image001.jpg>

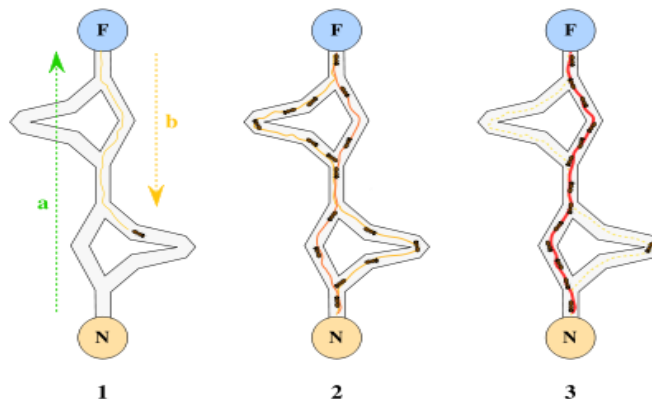
Schwarm-Simulation

Der Ameisen-Algorithmus

Lena-Sophie Schwabe und Elena Grygorova

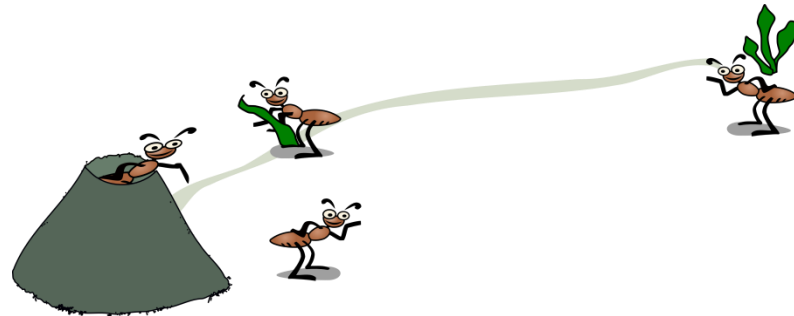
Problem

- ▶ Umsetzung des Ameisenalgorithmus
- ▶ Graph mit Knoten und Kanten eingelesen
- ▶ Ein Home-Knoten, ein Food-Knoten
- ▶ n Ameisen auf beliebigen Knoten platziert
- ▶ Starten und suchen als erstes die Futterquelle
- ▶ Wenn gefunden \rightarrow wieder umkehren und zum Home



Problem

- ▶ Dabei lassen die Ameisen Duftstoffe (Pheromonen) auf den Kanten
- ▶ diese verdunsten mit der Zeit
- ▶ Auf den kürzesten Wegen liegen irgendwann die meisten Duftstoffe → Kommunikation
- ▶ Wahrscheinlichkeit, dass die Ameise einen bestimmten Pfad einschlägt, hängt sehr von den Pheromonenwerten des Pfads ab



Parallelisierung

- ▶ OpenMP
- ▶ Im parallelen Abschnitt erstellt nur ein Thread verschiedene Tasks für die anderen Threads
- ▶ Es werden so viele Tasks angelegt wie es Ameisen gibt
- ▶ Die `startLife()`-Methode macht so lange Ameisenschritte, bis die Lebensdauer der Kolonie zu Ende geht
- ▶ In der `step()`-Methode wird die Wahrscheinlichkeit des nächsten Weges mit einer Formel berechnet
- ▶ Es wird außerdem getestet, ob das Ziel schon erreicht wurde



Parallelisierung

- ▶ Es wird ebenfalls ein Task für das Update der Pheromonenwerte erstellt
- ▶ Die update()-Funktion hat die Dauer der Lebenszeit der Kolonie (Eingabeparameter)
- ▶ Die Aktualisierung findet in Intervallen statt (ebenfalls Eingabeparameter)



MPI

- ▶ Jeder Rank hat einen eigenen Graphen
- ▶ Beim Aufruf der `update()`-Funktion → der neue Pheromonenwert wird an alle Ranks geschickt
- ▶ Der Durchschnitt von allen Pheromonenwerten wird gebildet, damit alle Ranks am Ende das gleiche haben



Problem

- ▶ Bei einigen Ranks war die Zeit bereits abgelaufen und die `update()`-Funktion wurde verlassen
- ▶ Andere Ranks haben damit vergeblich auf die Nachricht gewartet
- ▶ Lösung: den Integer status eingeführt
- ▶ Am Anfang ist er 0, wenn die Zeit abläuft, wird er auf 1 gesetzt
- ▶ Status wird ebenfalls mit `MPI_Reduce()` hin- und hergeschickt
- ▶ Wenn status 1 beträgt, passiert nur noch 1 Update und die Ranks verlassen die Funktion



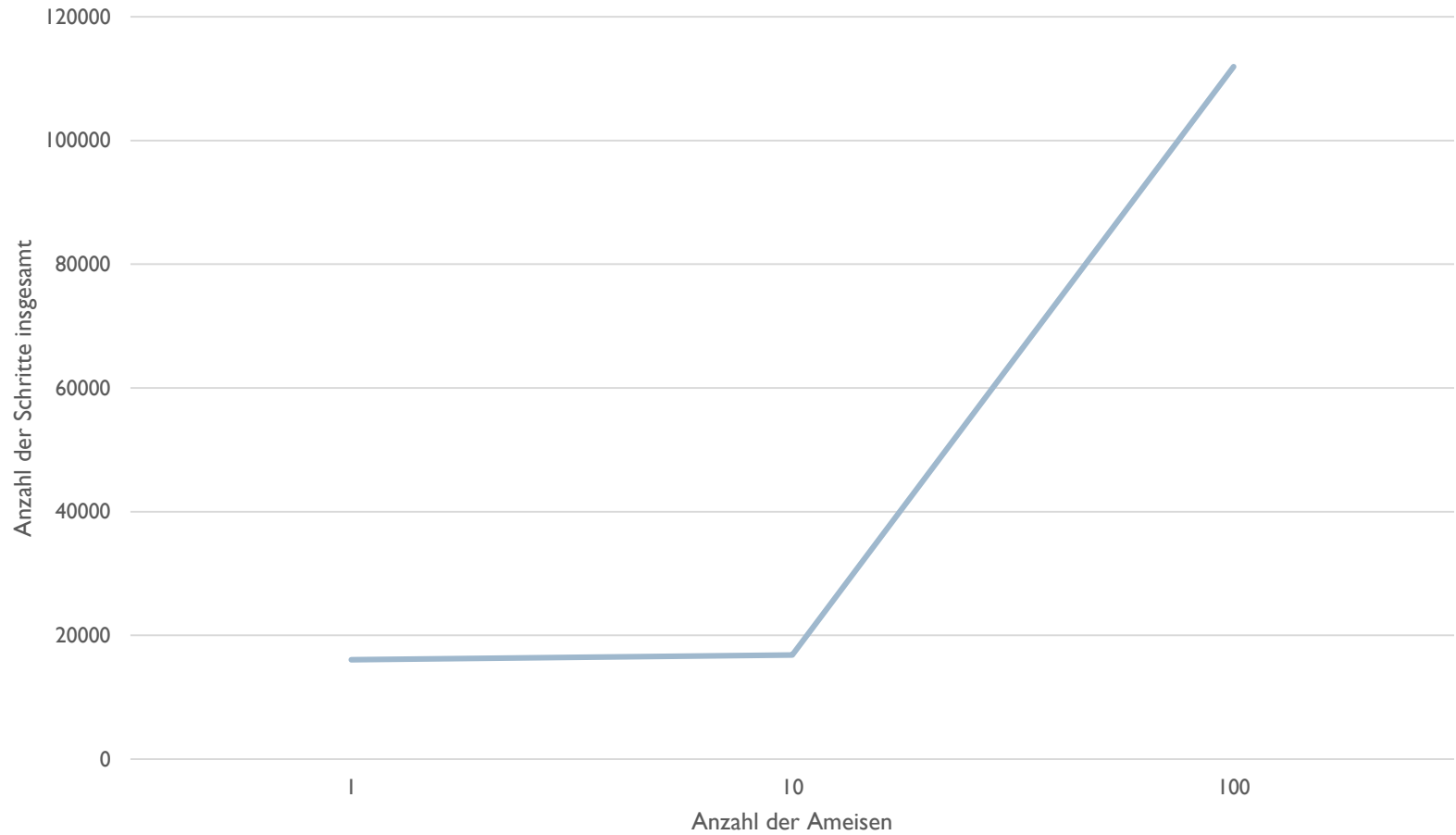
Problem

- ▶ Ursprüngliche Idee: Kantenmatrix → Knoten x Knoten
→ zu groß!
- ▶ Umwandlung in `std::unordered_map<std::string, edge_struct>`
- ▶ String: `start_id : end_id` → key
- ▶ `Edge_struct` enthält Pheromonenwert



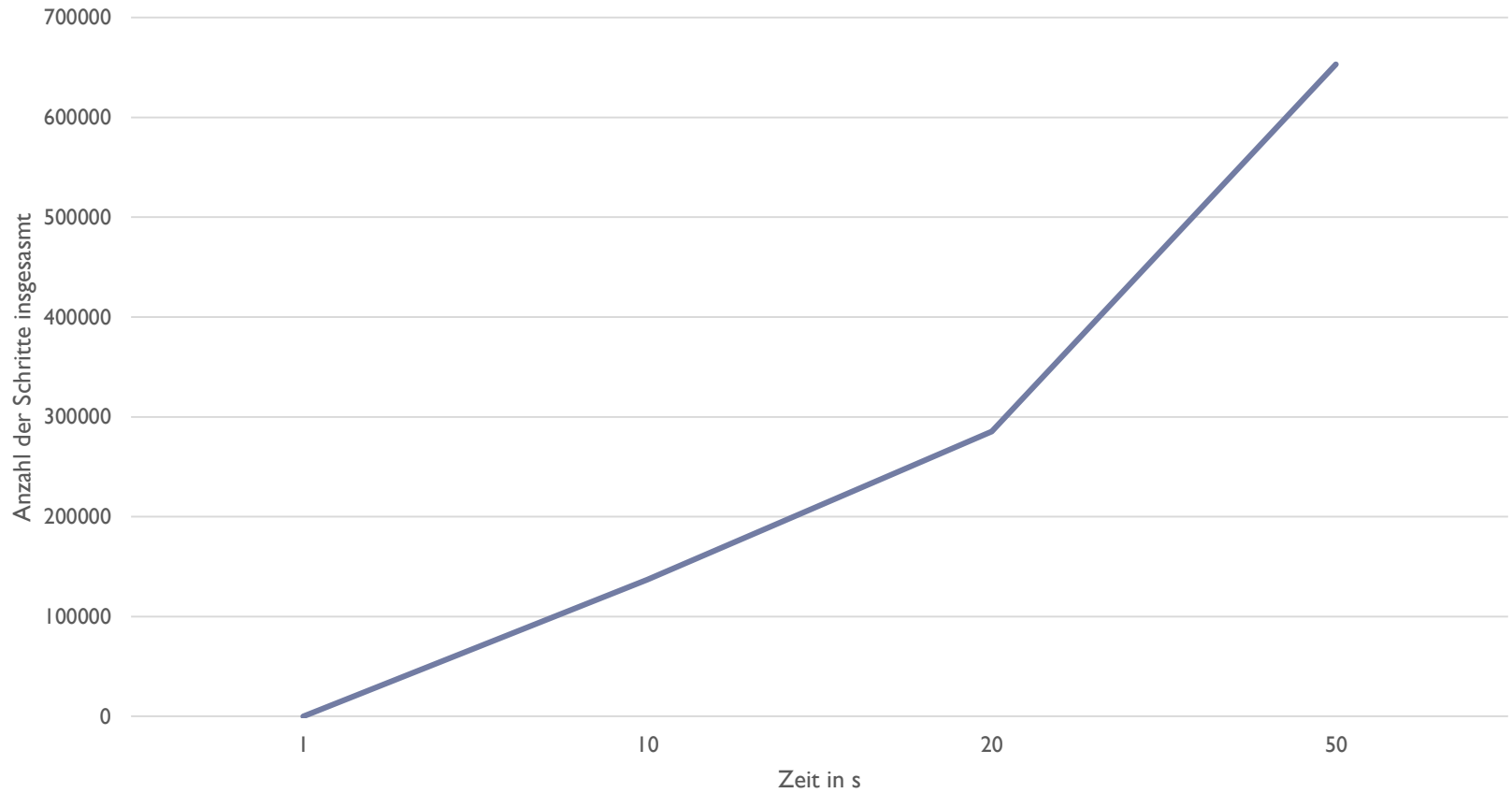
Zeitmessungen und Speed-Up

Chart Title



Zeitmessungen und Speed-Up

Chart Title



Demo des Programms an einem Beispiel-Graph

- ▶ Graph mit 50 Knoten und 85 Kanten
- ▶ 100 Ameisen als Beispiel
- ▶ 20 Sekunden Lebenszeit der Kolonie
- ▶ Kürzester Weg = 7

