



# Effiziente parallele Vernetzung hierarchischer Baugruppen auf HPC-Clustern

Bachelorarbeit, Ausschreibungsdatum: 8. November 2021

## Hintergrund

Der Lehrstuhl für Konstruktionslehre und CAD forscht an innovativen Themen u. a. in den Bereichen Simulation und FEM. Ein Vorbereitungsschritt für eine FEM-Simulation besteht darin, für ein zu simulierendes Bauteil eine Netzstruktur zu bestimmen, welche das Bauteil in hinreichend kleine Elemente zerlegt. Für diesen Schritt existiert bereits Standardsoftware, z. B. gmsh [1].

Der Trend bei der Simulation von Bauteilen und Prozessen geht hin zu immer größeren und komplexeren Modellen, die immer genauer die Realität abbilden. Betrachtet werden soll in dieser Arbeit die Vernetzung komplexer Baugruppen (z. B. Satelliten), welche aus einer hierarchischen Struktur von Unterbaugruppen (z. B. Rumpf, Solarpanel) und Bauteilen (z. B. Träger, Schrauben) bestehen. Unterschiedlich große und komplexe Bauteile führen dabei zu unterschiedlichem individuellem Rechenaufwand. Es soll der Zeitaufwand für die Vernetzung komplexer hierarchischer Baugruppen reduziert werden, indem eine effiziente parallele Berechnung auf HPC-Clustern erreicht wird. Dabei ist insbesondere der unterschiedliche Rechenaufwand pro Bauteil zu beachten.

## Aufgabenstellung

Am Lehrstuhl für Konstruktionslehre und CAD wurden bereits Cluster-Scripte entwickelt, welche eine parallele Vernetzung durch Anwendung von gmsh auf die einzelnen Bauteile prinzipiell ermöglichen, jedoch noch keine automatische Lastbalancierung beherrschen. Ziel der ausgeschriebenen Arbeit ist es, den vorhandenen Workflow weiterzuentwickeln und insbesondere Möglichkeiten zur automatischen Lastbalancierung zu untersuchen und zu integrieren. Für die Vernetzung der einzelnen Bauteile soll weiterhin Standardsoftware (z. B. gmsh) verwendet werden.

In einem ersten Schritt soll davon ausgegangen werden, dass wesentlich mehr Bauteile vorhanden sind, als CPU-Ressourcen verwendet werden sollen. Die Vernetzung der einzelnen Bauteile soll in diesem Fall durch sequentielle Programmläufe der Vernetzungssoftware erfolgen, wobei die zu erstellenden Steuer-Scripte für eine dynamische Lastverteilung der sequentiellen Programme auf die verfügbaren CPU-Ressourcen sorgen.

Da einige Vernetzungsprogramme (u. a. gmsh) selbst parallelisiert sind, soll in einem zweiten Schritt untersucht werden, inwieweit sich durch Ausnutzung dieses zusätzlichen Parallelitätsgrades die Zahl der nutzbaren CPU-Ressourcen vergrößern und die Gesamtzeit für die Vernetzung dadurch reduzieren lässt. Im Grenzfall wäre denkbar, alle Bauteile gleichzeitig zu berechnen und dabei die Parallelität des Vernetzungsprogramms für jedes Bauteil so zu wählen, dass alle Berechnungen gleichzeitig enden. Dazu könnte in erster Näherung das vorab zu bestimmende Bauteilvolumen als Kriterium herangezogen werden.

Der Erfolg der umgesetzten Lösung(en) ist durch Laufzeit- und Skalierbarkeitsuntersuchungen experimentell zu bestätigen. Insbesondere für den zweiten Schritt bilden solche Untersuchungen die Grundlage für die Zuordnungen zwischen Bauteilen und CPU-Ressourcen.

## Erforderliche Kenntnisse und Fähigkeiten

Linux, Script-Programmierung (Python, Shell), Konzepte von MPI und Threads, Laufzeit- und Skalierbarkeitsanalyse paralleler Programme

## Betreuer

Christopher Lange<sup>1</sup>, Prof. Dr. Matthias Korch<sup>2</sup>

## Literatur

[1] C. Geuzaine and J.-F. Remacle. *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. International Journal for Numerical Methods in Engineering 79(11), pp. 1309–1331, 2009. See also <https://gmsh.info>.

<sup>1</sup> Lehrstuhl Konstruktionslehre und CAD, Büro: FAN C 1.38, Tel.: 7180, Email: [christopher.lange@uni-bayreuth.de](mailto:christopher.lange@uni-bayreuth.de)

<sup>2</sup> Büro: AI 2.11, Tel.: 7705, E-Mail: [korch@uni-bayreuth.de](mailto:korch@uni-bayreuth.de)