
Kurzfassung

Ziel dieser Arbeit war die Beschleunigung des von Elena-Ruxandra Cojocaru und Sébastien Bérújon in Python implementierten Wellenfrontrekonstruktionsalgorithmus. Dieser berechnet aus zwei Bildern einer Probe pixelweise die Fronten der elektromagnetischen Welle eines Röntgenlasers. Die Bilder werden hierbei von zwei hochempfindlichen Röntgen-CCD-Sensoren aufgenommen, welche in einem festen Abstand zueinander und zur Probe positioniert sind. Treffen Strahlen des Röntgenlasers auf diese, so lässt sich aus den so aufgenommenen Streubildern die Wellenfront rekonstruieren, was Rückschlüsse auf die Struktur der Probe zulässt. Auf Basis von Performance-Analysen der Python-Implementierung wurden Optimierungen und Parallelisierungsmöglichkeiten für die kritischen Programmabschnitte ermittelt, implementiert und evaluiert. Die schnellste vorgestellte Lösung basiert auf der Verteilung der Bildpaare auf mehrere Rechenkerngruppen und der Parallelisierung der Berechnung der Bildpaare auf diesen, was eine Skalierung über mehrere Knoten erlaubt. Kombiniert mit der Nutzung optimierter Bibliotheken und dem Übersetzen des Python-Codes wurde eine Beschleunigung von bis zu vier gegenüber der Referenzimplementierung mit gleicher Kernanzahl erreicht. Wurden 120 Kerne verwendet, so war eine Beschleunigung auf das bis zu 133-fache gegenüber der Referenz auf einem Kern möglich. Die Referenzdaten hierfür wurden an der Beamline BM05 der European Synchrotron Radiation Facility aufgenommen.

Abstract

The goal of this thesis was the acceleration of the wavefront reconstruction algorithm which was developed in Python by Elena-Ruxandra Cojocaru and Sébastien Bérújon. This algorithm calculates the electromagnetic wavefront of an X-ray laser from two images of a target pixelwise. The images were captured by two highly sensitive X-ray CCD sensors, which were positioned in a fixed distance to each other and the target. When the refracted X-ray beam hits these detectors a distortion image is generated from which the wavefront can be reconstructed. The result can be used to draw conclusions about the structure of the target. On the basis of performance measurements of the Python implementation optimization and parallelization possibilities for critical sections were determined, implemented and evaluated. The fastest proposed solution is based on the distribution of the image pairs onto CPU core groups and the parallelization of the calculation of the image pairs on these which allows scaling the problem over multiple nodes. This combined with the use of optimized libraries and the compilation of the Python code resulted in a speedup of up to four towards the reference implementation without the use of more cores. When using 120 cores a speedup of up to 133 towards the reference implementation running on a single core could be achieved. The here used datasets were recorded at Beamline BM05 of the European Synchrotron Radiation Facility.