

Parallelisierung von bewegten Teilchen in unstrukturierten Gittern

Master-Arbeit

Einleitung

In Teilchenbeschleunigern werden Elektronen, Protonen oder Ionen in Bündeln zu hohen Energien beschleunigt. Elektromagnetische Felder und Teilchen und Teilchen wirken in komplizierter Weise aufeinander. Das elektromagnetische Feld bewegt die Teilchen. Andererseits beeinflusst die Dynamik der Teilchen wiederum das Feld. In einem Teilchenbündel befinden sich sehr viele (mehr als 10^9) Teilchen. Einerseits stossen sich die Teilchen gegenseitig ab (Coulombkraft) andererseits gibt es aber auch fokussierende und ablenkende Kräfte welche die Teilchen auf ihrer Bahn halten. Durch diese Einflüsse entstehen sehr komplexe Vorgänge. Die Physik der Beschleuniger ist die Elektrodynamik. Mathematisch müssen partielle Differentialgleichungen gelöst werden (Maxwell Gleichungen).

Die Simulation von Teilchenbeschleunigern ist zu einem wichtigen Werkzeug beim Entwerfen und in der Wartung von solch faszinierenden Maschinen geworden. Um genaue Angaben über die Beschleuniger-Physik machen zu können, ist es unabdingbar das Plasma angemessen zu modellieren – d.h. es müssen sehr viele Teilchen in die Berechnung einbezogen werden. Da dies auf einem einzigen Prozessor wegen mangelnder Speicherkapazität nicht möglich ist, muss das Problem parallel gelöst werden.

Aufgabestellung

Zur Verfügung steht ein am Paul Scherrer Institut entwickeltes Framework (IPPL [1]), das die Teilchen parallel behandelt. Tests haben ergeben, dass IPPL bis mindestens 8000 Prozessoren skaliert. In diesem Framework sind die MPI Befehle in Methoden einer höheren Abstraktionsebene gekapselt – das heisst es muss nicht direkt mit MPI interagiert werden.

Die Teilchen-Klasse nimmt als Templateparameter ein Objekt vom Typ `ParticleLayout` an. `ParticleLayout` ist für die Parallelisierung zuständig, das heisst, es kennt die Zugehörigkeit der Teilchen zu den Prozessoren anhand eines partitionierten Gitters (siehe Abbildung). Nach jedem Zeitschritt werden die Teilchen durch einen sog. *particle pusher* gemäss den auf sie wirkenden Kräften verschoben. Falls ein Teilchen in eine Zelle wechselt, die auf einem anderen Prozessor liegt muss dies den jeweiligen Prozessoren mitgeteilt werden (Kommunikation), dass diese ihre Elementtabelle nachführen können.

Das Framework soll so erweitert werden, dass auch unstrukturierte Gitter (mit Tetraeder- und Hexaederelementen) benutzt werden können. Unstrukturierte Gitter können oft nicht einfach in

Teilgebiete gleicher Grösse aufgeteilt werden. Zudem ist die Kommunikation weniger ausgeglichen als bei Rechtecksgittern.

In dieser Arbeit wird das 3-dimensionale Gitter durch ein vorhandenes Programm erzeugt (Netgen [2]). Im Hinblick auf optimale Lastverteilung und minimales Kommunikationsvolumen soll das Gitter mit Hilfe von ParMetis [3] oder Scotch [4] partitioniert werden.

Zusätzlich zur Anpassung von `ParticleLayout` sollen Überlegungen angestellt werden, wie Teilchen im Kontext von Finiten Elementen (für die Berechnung der elektromagnetischen Felder) effizient parallelisiert werden können.

Die Rechnungen, die viele Ressourcen benötigen, werden auf dem Opteron-Cluster *Brutus* [5] der ETH ausgeführt.

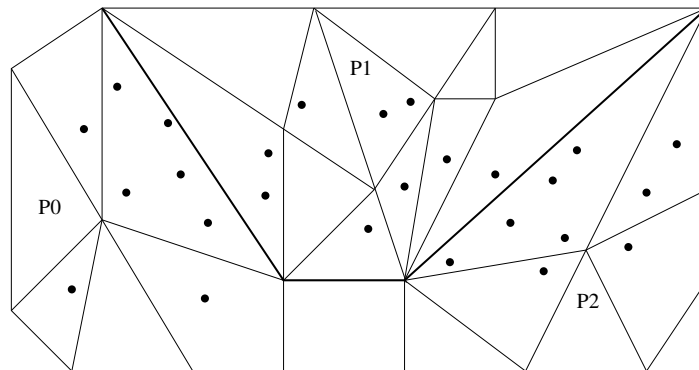


Abbildung 1: Unstrukturiertes Gitter mit Teilchen (aufgeteilt auf 3 Prozessoren)

Anforderungen

- Fundierte Programmierkenntnisse in C++ und OOP Verständnis
- Es ist von Vorteil, wenn die Vorlesung “Paralleles Rechnen” besucht wurde

Kontakt

- PSI: Andreas Adelman, andreas.adelman@psi.ch, Tel: 056 310 42 33
- ETHZ: Marcus Wittberger, marcus.wittberger@inf.ethz.ch, Tel: 044 632 75 47

Literatur

- [1] Independent Parallel Particle Layer: <http://amas.web.psi.ch>
- [2] Netgen: <http://www.hp fem.jku.at/index.html?joachim/>
- [3] ParMetis: <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview/>
- [4] Scotch: <http://www.labri.fr/perso/pelegrin/scotch/>
- [5] Brutus Cluster: <http://clusterwiki.ethz.ch/wiki/index.php/Brutus>

26. 6. 2008 P. Arbenz/A. Adelman/M. Wittberger