Dokumentation – ELLPACK Format

Die Matrix-Vektor-Multiplikation besitzt inhärent eine niedrige arithmetische Dichte, denn pro geladenem Matrixwert wird nur eine Rechenoperation (ein Multiply-Add) ausgeführt. Für eine gute Performance auf Grafikkarten ist es daher wichtig, Daten aus dem Hauptspeicher möglichst effizient zu laden. Die Memory Controller können 32-, 64- und 128-Byte Transaktionen ausführen, dabei aber jeweils nur an durch die Größe teilbaren Adressen beginnen. Dementsprechend sollte ein Kernel

* Speicheroperationen zu größeren Transaktionen zusammenfassen können (coalesced memory access). Dazu müssen aufeinanderfolgende Threads auf aufeinanderfolgende Adressen im Speicher zugreifen.
* Transaktionen möglichst an der benötigten Adresse beginnen (aligned memory access).

Um die volle Speicherbandbreite zu nutzen, müssen alle 16 Threads eines Half Warps an einer 128-Byte-aligned Adresse beginnend jeweils aufeinanderfolgende (8-Byte-)Double-Werte aus der Matrix-Datenstruktur lesen, die dann zu einer einzigen 128-Byte Transaktion zusammengefasst werden.

Für diesen Zweck wurde das ELLPACK Format entwickelt, das die Matrix entsprechend dieser Voraussetzung in zwei Datenstrukturen ablegt, eine für die Werte und eine für die Spaltenindizes.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | a | b |  |  |  |  |  | c | |  |  |  | d | e |  |  |  | | f |  |  |  |  | g | h |  | |  | i | j | k |  |  |  |  | |  |  |  |  |  | l | m |  | |  |  |  | n |  |  |  |  | |  | |  |  |  | | --- | --- | --- | | a | b | c | | d | e | - | | f | g | h | | i | j | k | | l | m | - | | n | - | - | | |  |  |  | | --- | --- | --- | | 0 | 1 | 7 | | 3 | 4 | - | | 0 | 5 | 6 | | 1 | 2 | 3 | | 5 | 6 | - | | 3 | - | - | | Block 0  Block 1 |
| ursprüngliche Matrix |  | Matrixwerte | Spaltenindizes |  |
|  |  |  |  |  |
| Matrixwerte: | [ a d f b e g c - h i l n j m - k - - ] | | | |
| Spaltenindizes: | [ 0 3 0 1 4 5 7 - 6 1 5 3 2 6 - 3 - - ] | | | |

Die Matrix wird zeilenweise in Blöcke aufgeteilt, wobei jeder Thread eine Zeile übernimmt. Dann werden die Matrixwerte und Spaltenindizes pro Block spaltenweise abgespeichert, sodass im Feld erst nacheinander alle Spalten von Block 0, dann alle von Block 1 und so weiter stehen. Die Spalten werden so aufgefüllt, dass alle gleich lang sind, sprich auf die Länge der längsten Zeile. Damit der Overhead nicht zu groß wird, gehen wir davon aus, dass alle Zeilen in etwa gleich lang sind.

Im Algorithmus geht nun jeder Thread in einer Schleife durch seine Zeile und multipliziert den Matrixwert an jeder Stelle mit dem Wert des Vektors, dessen Index er aus dem Spaltenindex erhält. Daran erkennt man, dass alle Threads in einem Block in beiden Datenstrukturen der Matrix aufeinanderfolgende Werte lesen, die Speicheroperationen können also zusammengefasst werden. Fügt man nun noch ein Padding mit Leerspalten ein, damit die Zahl der Spalten immer ein Vielfaches von 16 ist, so fängt jeder Block und in jedem Block jeder Half-Warp an einer 128-Byte-aligned Adresse an.

Einzig die Werte des Vektors werden nicht organisiert geladen, denn deren Struktur ist durch die Spaltenindizes unregelmäßig und hängt von der konkreten Matrix ab. Benutzt man eine Grafikkarte mit Cache (Compute Capability 2.0 und höher), sollte dieser den Effekt allerdings abmildern.