

Safety-Critical Systems

Speicherkonsistenzmodelle fuer GPUs

Marc Blickle
Master Of Science - Informatik

Datum des Vortrags

Kurzfassung

Eine kurze Zusammenfassung der Ausarbeitung mit 10-12 Zeilen Text.

Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen	4
2.1	CPU und Arbeitsspeicher	4
2.1.1	Multicoreprozessoren und Shared-Memory	4
2.1.2	Speicherkonsistenz	5
2.2	Der Grafikprozessor (GPU)	6
2.2.1	Speicher in der GPU	7
2.2.2	Ist-Zustand, allgemeine Problemstellung	8
2.2.3	Anforderungen	9
3	Speicherkonsistenzmodelle - Gegenüberstellung	9
3.1	Strict Consistency Model	10
3.1.1	Beschreibung	10
3.1.2	Vor- und Nachteile	10
3.2	Sequential Consistency Model	11
3.2.1	Beschreibung	11
3.2.2	Vor- und Nachteile	11
3.3	Weak Consistency Model	12
3.3.1	Beschreibung	12
3.3.2	Vor- und Nachteile	12
3.4	Cache Consistency Model	13
3.4.1	Beschreibung	13
3.4.2	Vor- und Nachteile	13
3.5	Slow Consistency Model	14
3.5.1	Beschreibung	14
3.5.2	Vor- und Nachteile	14
3.6	PRAM Consistency Model	15
3.6.1	Beschreibung	15
3.6.2	Vor- und Nachteile	15
3.7	Processor Consistency Model	16
3.7.1	Beschreibung	16
3.7.2	Vor- und Nachteile	16
3.8	Release Consistency Model	17
3.8.1	Beschreibung	17
3.8.2	Vor- und Nachteile	17
3.9	Evaluation und Gegenüberstellung	18
4	Zusammenfassung und Fazit	19
5	Literatur	20

1 Einleitung

Prozessoreinheiten mit mehreren Kernen finden schon seit etlichen Jahren Anwendung in Systemen jeder Art: Vom Mobiltelefon, über heimische Desktop-PCs bis hin zu mächtigen Serverkonstellationen. Um diesen Prozessoren ein reibungsloses Zusammenspiel mit dem Arbeitsspeicher zu garantieren ist es von hoher Wichtigkeit, die sich im Verarbeitungsprozess befindlichen Dateien konsistent zu halten. Dies erreicht man mitunter durch die Entwicklung und Implementation von sogenannten Speicherkonsistenz-Modellen. Mittlerweile bewegt sich der Trend bei der Applikationsentwicklung aber vermehrt in Richtung paralleler Datenverarbeitung mithilfe von GPUs, wie zum Beispiel beim Minen von Bitcoins.

Ein großer Mangel stellt hierbei noch das Verwenden entsprechender Speicherkonsistenzmodelle dar, da das bisherige Anwendungsgebiet von GPUs, nämlich grafisch anspruchsvolle Applikationen, eine höhere Fehlertoleranz hat. In den meisten Grafikchips kommen nur sehr verzeihende und schwach Modelle zum Einsatz oder sie sind nur sehr dürftig beschrieben. Dies erschwert es für Entwickler ungemein, neue Applikationen zu verwirklichen.

In dieser Arbeit werden deshalb verschiedene Speicherkonsistenzmodelle, die sich beim Einsatz in CPUs bewährt haben, gegenübergestellt und bezüglich den Anforderungen in GPUs evaluiert.

2 Grundlagen

Quellen sind unten aufgeführt

2.1 CPU und Arbeitsspeicher

-Kurze Beschreibung Zusammenarbeit CPU und RAM

2.1.1 Multicoreprozessoren und Shared-Memory

-Beschreibung Zusammenarbeit mehrere Kerne und RAM

2.1.2 Speicherkonsistenz

Was bedeutet Konsistenz? Was ist das Problem bei Sharedmemory usw Wie kann es gelöst werden Was sind Modelle (allgemein), was ist ihre Aufgabe..

2.2 Der Grafikprozessor (GPU)

2.2.1 Speicher in der GPU

Was ist der Unterschied zu CPU? Warum brauchen wir Konsistenzmodelle?

2.2.2 Ist-Zustand, allgemeine Problemstellung

-Rudimentäre Modelle -Wenig Info kommt von Herstellern -Schwere Applikationsentwicklung

2.2.3 Anforderungen

Ausarbeitung der Anforderungen

3 Speicherkonsistenzmodelle - Gegenüberstellung

Vorstellen verschiedener Modelle, Vor-und Nachteile

3.1 Strict Consistency Model

3.1.1 Beschreibung

-Besonderheiten und Funktionsweise des Modells

3.1.2 Vor- und Nachteile

-Was macht dieses Modell gut? -Was macht dieses Modell schlecht? -Warum ist das gut oder schlecht für Grafikchips? Quellen: Die Informationen werden aus den unten aufgeführten Quellen zusammengetragen, sie alle beschreiben unterschiedliche Aspekte des Modells

3.2 Sequential Consistency Model

3.2.1 Beschreibung

-Besonderheiten und Funktionsweise des Modells

3.2.2 Vor- und Nachteile

-Was macht dieses Modell gut? -Was macht dieses Modell schlecht? -Warum ist das gut oder schlecht für Grafikchips? Quellen: Die Informationen werden aus den unten aufgeführten Quellen zusammengetragen, sie alle beschreiben unterschiedliche Aspekte des Modells

3.3 Weak Consistency Model

3.3.1 Beschreibung

-Besonderheiten und Funktionsweise des Modells

3.3.2 Vor- und Nachteile

-Was macht dieses Modell gut? -Was macht dieses Modell schlecht? -Warum ist das gut oder schlecht für Grafikchips? Quellen: Die Informationen werden aus den unten aufgeführten Quellen zusammengetragen, sie alle beschreiben unterschiedliche Aspekte des Modells

3.4 Cache Consistency Model

3.4.1 Beschreibung

-Besonderheiten und Funktionsweise des Modells

3.4.2 Vor- und Nachteile

-Was macht dieses Modell gut? -Was macht dieses Modell schlecht? -Warum ist das gut oder schlecht für Grafikchips? Quellen: Die Informationen werden aus den unten aufgeführten Quellen zusammengetragen, sie alle beschreiben unterschiedliche Aspekte des Modells

3.5 Slow Consistency Model

3.5.1 Beschreibung

-Besonderheiten und Funktionsweise des Modells

3.5.2 Vor- und Nachteile

-Was macht dieses Modell gut? -Was macht dieses Modell schlecht? -Warum ist das gut oder schlecht für Grafikchips? Quellen: Die Informationen werden aus den unten aufgeführten Quellen zusammengetragen, sie alle beschreiben unterschiedliche Aspekte des Modells

3.6 PRAM Consistency Model

3.6.1 Beschreibung

-Besonderheiten und Funktionsweise des Modells

3.6.2 Vor- und Nachteile

-Was macht dieses Modell gut? -Was macht dieses Modell schlecht? -Warum ist das gut oder schlecht für Grafikchips? Quellen: Die Informationen werden aus den unten aufgeführten Quellen zusammengetragen, sie alle beschreiben unterschiedliche Aspekte des Modells

3.7 Processor Consistency Model

3.7.1 Beschreibung

-Besonderheiten und Funktionsweise des Modells

3.7.2 Vor- und Nachteile

-Was macht dieses Modell gut? -Was macht dieses Modell schlecht? -Warum ist das gut oder schlecht für Grafikchips? Quellen: Die Informationen werden aus den unten aufgeführten Quellen zusammengetragen, sie alle beschreiben unterschiedliche Aspekte des Modells

3.8 Release Consistency Model

3.8.1 Beschreibung

-Besonderheiten und Funktionsweise des Modells

3.8.2 Vor- und Nachteile

-Was macht dieses Modell gut? -Was macht dieses Modell schlecht? -Warum ist das gut oder schlecht für Grafikchips? Quellen: Die Informationen werden aus den unten aufgeführten Quellen zusammengetragen, sie alle beschreiben unterschiedliche Aspekte des Modells

3.9 Evaluation und Gegenüberstellung

-Die zuvor erarbeiteten Vor und Nachteile werden hier Gegenübergestellt und ein Fazit gezogen -Gewichtungstabelle

4 Zusammenfassung und Fazit

5 Literatur

- [Adv93] Sarita V. Adve. “Designing Memory Consistency Models For Shared-Memory Multiprocessors”. In: *University of Wisconsin-Madison* (Dec. 1993).
- [AG96] Sarita V. Adve and Kourosh Gharachorloo. “Shared Memory Consistency Models”. In: *Digital Equipment Corporation* (Dec. 1996).
- [Alg+10] Jade Alglave et al. “Fences In Weak Memory Models”. In: *ICAV* (2010).
- [CA12] Linchuan Chen and Gagan Agrawal. “Optimizing MapReduce for GPUs with Effective Shared Memory Usage”. In: *The Ohio State University , Department of Computer Science and Engineering* (June 2012).
- [FX11] Wu-chun Feng and Shucaï Xiao. “To GPU Synchronize or Not GPU Synchronize?” In: *Virginia Tech* (2011).
- [GCK14] Radhika Gogia, Preeti Chhabra, and Rupa Kumari. “Consistency Models in Distributed Shared Memory Systems”. In: *Dronacharya College Of Engineering* (Sept. 2014).
- [Gha95] Kourosh Gharachorloo. “Memory Consistency Models for Shared-Memory Multiprocessors”. In: *Stanford University, Departments of Electrical Engineering and Computer Science* (Dec. 1995).
- [Hil98] Mark D. Hill. “Multiprocessors Should Support Simple Memory-Consistency Models”. In: *University of Wisconsin, Madison* (1998).
- [Jan+11] Byunghyun Jang et al. “Exploiting Memory Access Patterns to Improve Memory Performance in Data-Parallel Architectures”. In: (2011).
- [Kin+09] Volodymyr V. Kindratenko et al. “GPU Clusters for High-Performance Computing”. In: *University of Illinois* (2009).
- [Lee09] Kenneth S. Lee. “Characterization and Exploitation of GPU Memory Systems”. In: *Virginia Polytechnic Institute and State University* (2009).
- [Mos] David Mosberger. “Memory Consistency Models”. In: *Department of Computer Science - The University of Arizona* ().
- [Rie16] Dennis Sebastian Rieber. “GPU Concurrency and Consistency”. In: *Universitaet Heidelberg* (2016).
- [SAN12] Abhayendra Singh, Shaizeen Aga, and Satish Narayanasamy. “Efficiently Enforcing Strong Memory Ordering in GPUs”. In: *University of Michigan, Department of Computer Science and Engineering* (June 2012).
- [Sar+11] Vijay A. Saraswat et al. “A theory of memory models”. In: *PPoPP* (2011).
- [SBS13] Akshay Sharma, Deepak Basora, and Ankur Sharma. “Memory Consistency Models”. In: *Dronacharya College of Engineering, International Journal of Enhanced Research in Management and Computer Applications* (Sept. 2013).

- [Sen13] Maximilian Senftleben. “Operational Characterization of Weak Memory Consistency Models”. In: *Department of Computer Science - University of Kaiserslautern* (2013).
- [SHW11] D.J. Sorin, M.D. Hill, and D.A. Wood. “A Primer on Memory Consistency and Cache coherence”. In: *Synthesis Lectures on Computer Architecture* (2011).
- [SN04] Robert C. Steinke and Gary J. Nutt. “A Unified Theory of Shared Memory Consistency”. In: *University of Colorado at Boulder, Department of Computer Science and Engineering* (Sept. 2004).
- [Wic+16] John Wickerson et al. “Automatically Comparing Memory Consistency Models”. In: *Imperial College London, University of Kent* (Nov. 2016).