GPU Concurrency
Weak Behaviours and Programming Assumptions

Jade Alglave[1,2], Mark Batty[3], Alastair F. Donaldson[4], Ganesh Gopalakrishnan[5], Jeroen Ketema[4], Daniel Poetzl[6], Tyler Sorensen[2,5] and John Wickerson[4]

[1]Microsoft Research Cambridge, [2]University College London, [3]University of Cambridge, [4]Imperial College London, [5]University of Utah, [6]University of Oxford

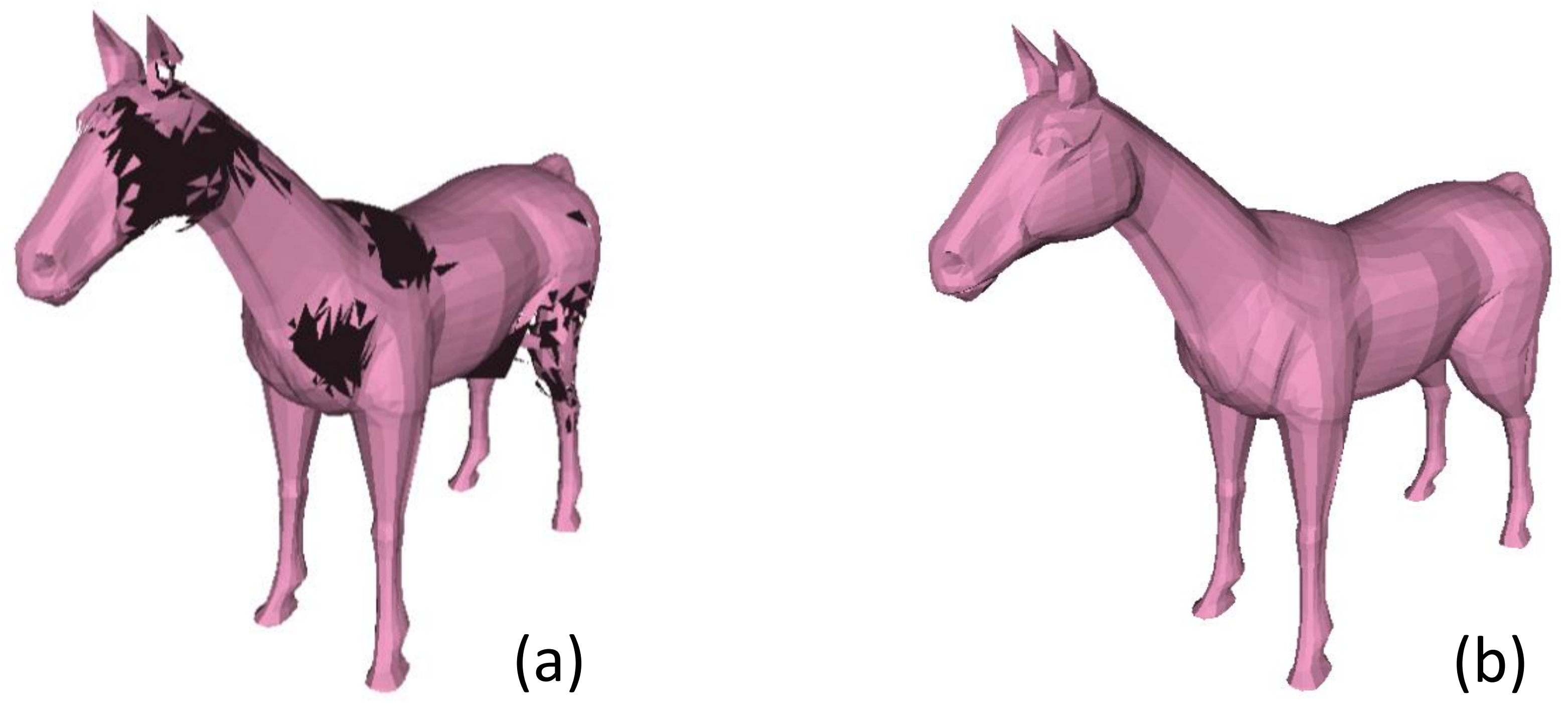
Motivation

- Multicore systems (e.g. Nvidia GPUs) implement *weak memory models* [1]*;* i.e. executions that do not correspond to an interleaving of concurrent instructions are observable.
- Documentation for such behaviours is often sparse and written in prose, which is prone to misinterpretations and can lead to bugs in applications.
- We explore which weak behaviours are experimentally observable on GPU chips; we compare our results to GPU applications containing synchronisation idioms. Finally, we give a formal GPU model which is sound w.r.t. our experimental data.

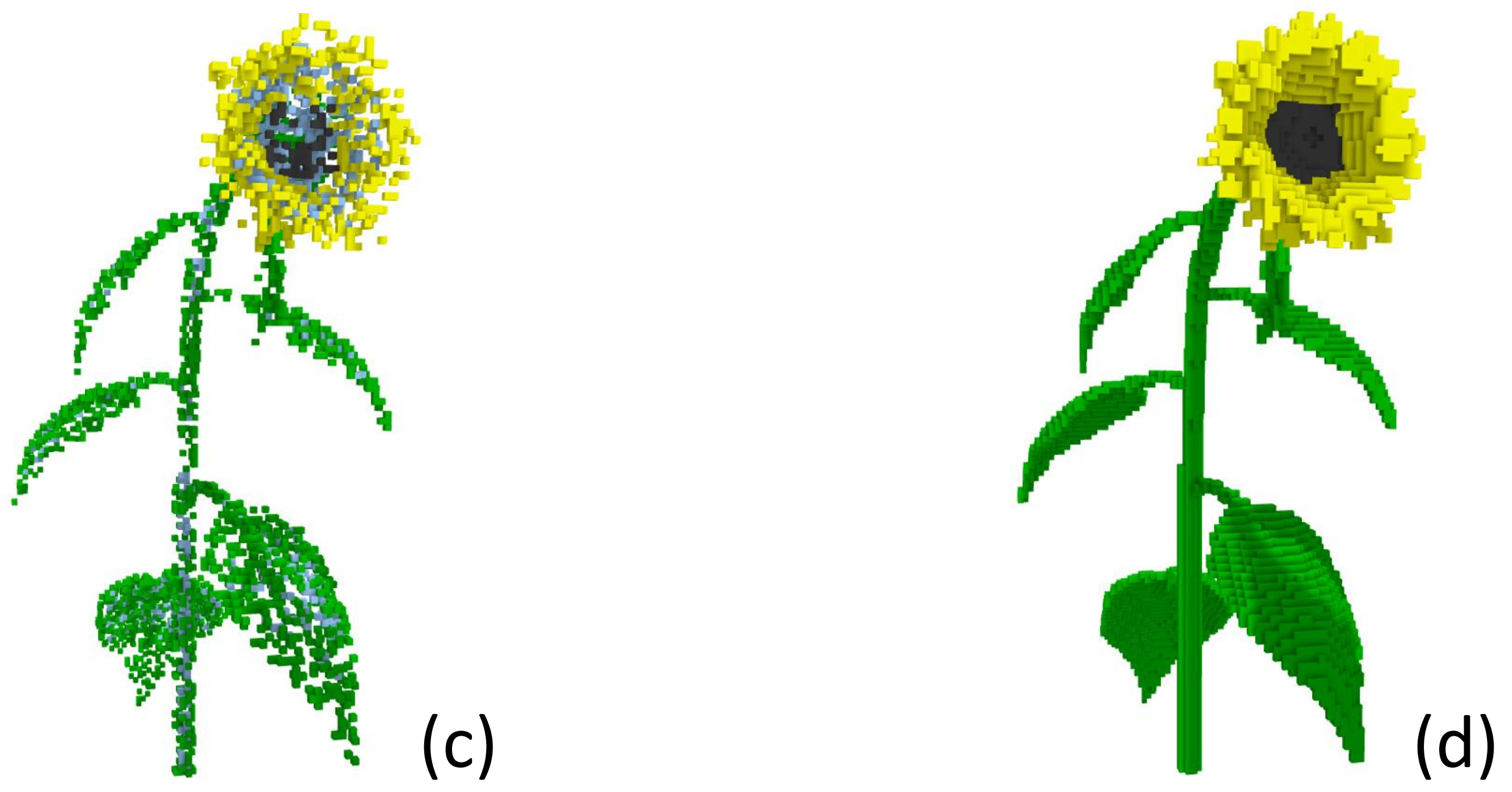
Methodology

		GPU additions
diy [1]	generates systematic families of litmus tests	concurrency and memory hierarchies (e.g. *scopes*)
litmus [1]	generates and executes code of a litmus test	stresses the system to increase the likelihood of weak behaviours
herd [1]	simulates a formal model given as a `cat` file	`cat` file for Nvidia PTX
targets	ARM, IBM PowerPC, and Intel x86 CPUs	AMD and Nvidia GPUs

Examples of observed behaviours



Pictures computed using an octree given in ***GPU Computing Gems: Jade Edition*** [2] on an Nvidia Tesla C2075. Errors in picture (a) are due to weak memory behaviours. Picture (b) is from code that has been experimentally fixed by us.



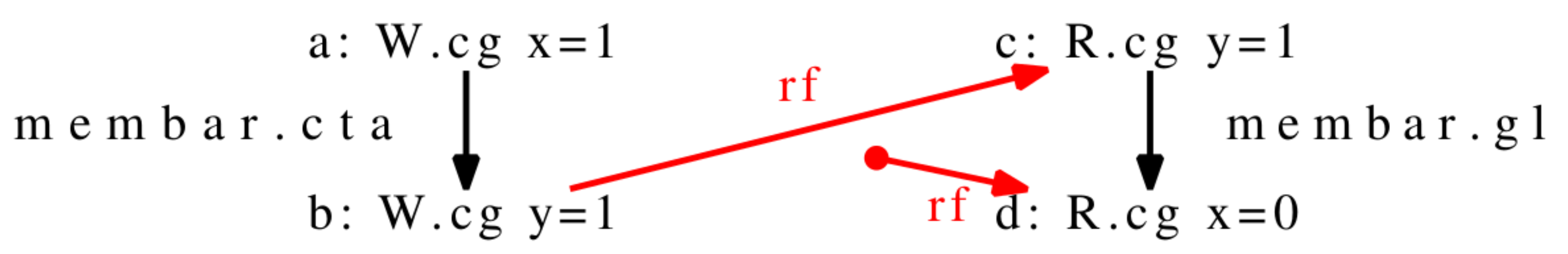
Pictures computed using a hash table in ***CUDA by Example*** [3] on an Nvidia Tesla C2075. Errors in picture (c) are due to weak memory behaviours. Picture (d) is from code that has been experimentally fixed by us. ***Led to an official Nvidia erratum*** [4]***.***

Formal model

We developed a formal model given as a `cat` file [1] for GPUs which is sound for over **10,000** litmus tests run on **5** Nvidia chips over **3** architectures (Fermi, Kepler, Maxwell)

init: $\begin{pmatrix}\texttt{global x=0}\\ \texttt{global y=0}\end{pmatrix}$ **final:** $\texttt{r0=1} \wedge \texttt{r2=0}$ **threads:** intra-CTA

```
a   st.cg [x],1           c   ld.cg r0,[y]
    membar.cta                 membar.gl
b   st.cg [y],1           d   ld.cg r2,[x]
```



Supplementary material

Paper: *GPU Concurrency: Weak Behaviours and Programming Assumptions*
Jade Alglave, Mark Batty, Alastair F. Donaldson, Ganesh Gopalakrishnan, Jeroen Ketema, Daniel Poetzl, Tyler Sorensen and John Wickerson.
ASPLOS '15.

Data:	http://virginia.cs.ucl.ac.uk/sunflowers/asplos15
Video:	http://youtu.be/3-Y8xLsqywY
Simulator:	http://virginia.cs.ucl.ac.uk/herd-web/?book=ptx

References:

[1] J. Alglave, L. Maranget, and M. Tautschnig. Herding cats: Modelling, simulation, testing, and data mining for weak memory. TOPLAS '14
[2] Wen-mei W. Hwu. GPU Computing Gems Jade Edition. Morgan Kaufmann Publishers Inc., 2011
[3] J. Sanders and E. Kandrot. CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison Wesley Professional, 2010
[4] https://developer.nvidia.com/cuda-example-errata-page

Support: EPSRC grants EP/{H005633, H008373, K008528, K011499, K039431}, EU FP7 project CARP (287767), NSF CCF Awards 1346756 and 1302449, and SRC project 2269.002