Efficient Correlation-Free Many-States Lattice Monte Carlo on GPUs

Jeffrey Kelling,
Géza Ódor, Martin Weigel, Sibylle Gemming

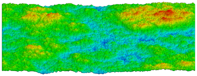
22nd June 2018

NVIDIA GPU CENTER OF EXCELLENCE

HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF NANONET

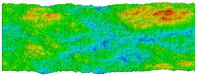
1 Introduction: What is this talk about?

- surface growth, physical aging (and non-equilibrium systems)

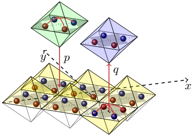


1 Introduction: What is this talk about?

- surface growth, physical aging (and non-equilibrium systems)

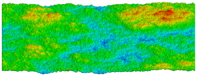


- lattice Monte-Carlo

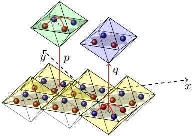


1 Introduction: What is this talk about?

- surface growth, physical aging (and non-equilibrium systems)



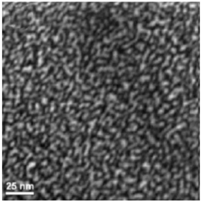
- lattice Monte-Carlo



2 Trivial parallism vs. SIMT

Applications for Monte Carlo: Stochastic Prosesses

http://hubblesite.org/newscenter/archive/releases/2007/17/image/a

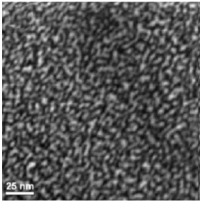


Müller, T., Heinig, K.-H. et al. *Appl. Phys. Lett.* **85** 2373 (2004)

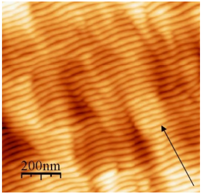
Applications for Monte Carlo: Stochastic Prosesses

http://hubblesite.org/newscenter/archive/releases/2007/17/image/a

http://en.wikipedia.org/wiki/File:Rub_al_Khali_002.JPG



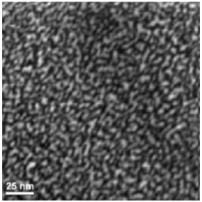
Müller, T., Heinig, K.-H. et al. *Appl. Phys. Lett.* **85** 2373 (2004)



https://www.hzdr.de/db/Cms?pOid=24344&pNid=2707

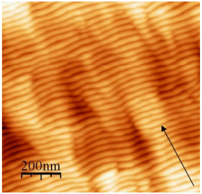
Applications for Monte Carlo: Stochastic Prosesses

http://hubblesite.org/newscenter/archive/releases/2007/17/image/a



Müller, T., Heinig, K.-H. et al. *Appl. Phys. Lett.* **85** 2373 (2004)

http://en.wikipedia.org/wiki/File:Rub_al_Khali_002.JPG



https://www.hzdr.de/db/Cms?pOid=24344&pNid=2707

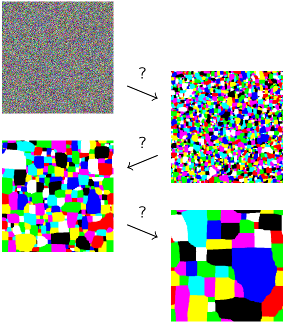
- game theory

  e. g.: Perc, Matjaž *Eur. J. Phys.* **38**(4) 045801 (2017)

  - sociology
  - finance
  - ...

Non-Equilibrium vs Equilibrium

out-of-Equilibrium:
kinetics of interest

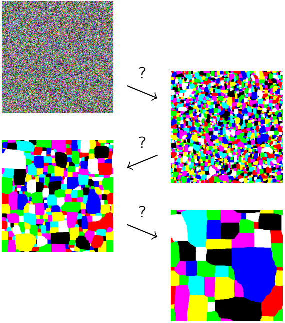


8-states Potts model, $\frac{J}{k_B T} = 5$

- optimal algorithm reproduces physical evolution

Non-Equilibrium vs Equilibrium

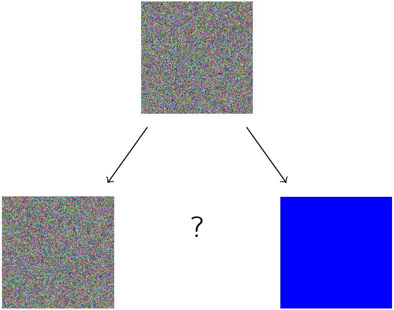
out-of-Equilibrium: **kinetics** of interest



8-states Potts model, $\frac{J}{k_B T} = 5$

- optimal algorithm reproduces physical evolution

Equilibrium Properties: only **final** state relevant



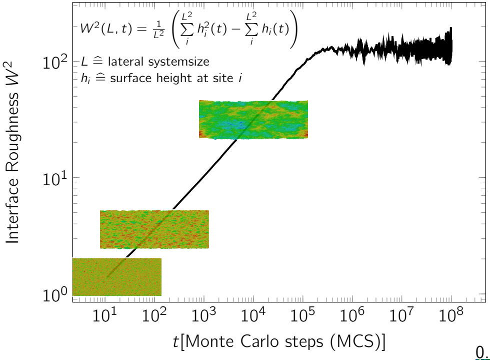
disordered state

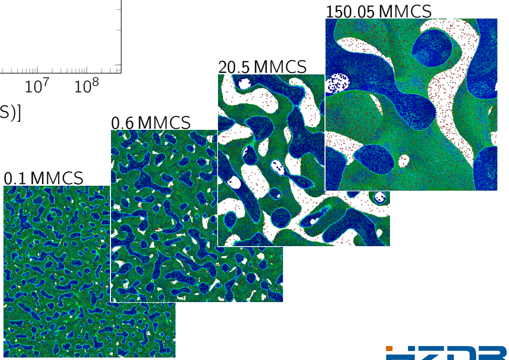
ordered state

8-states Potts model

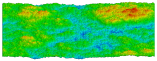
- optimal algorithm reaches equilibrium quickly

Non-Equilibrium Systems

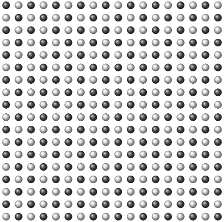




Domain Decomposition



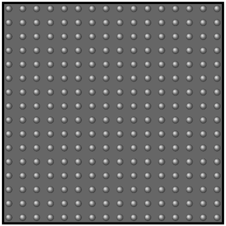
Stochastic Cellular Automaton (SCA)



- update odd/even sublattice
  update probability $p < 1$
- $+$ linear memory access $\Rightarrow$ fast

Kelling, J., Ódor, G., Gemming, S. *IEEE, INES Proc.* (2016)

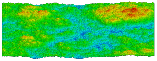
Random Sequential (RS)



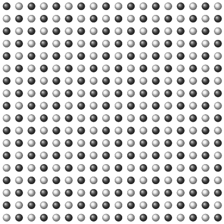
- $+$ uncorrelated updates

Kelling, J., Ódor, G., Gemming, S. *Comp. Phys. Commun.* **220** 205 (2017)
Kelling, J., Ódor, G., et al. *EPJST* **89** 175 (2012)

Domain Decomposition



Stochastic Cellular Automaton (SCA)

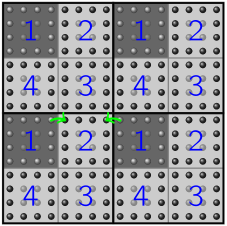


- update odd/even sublattice
  update probability $p < 1$

+ linear memory access ⇒ fast

Kelling, J., Ódor, G., Gemming, S. *IEEE, INES Proc.* (2016)

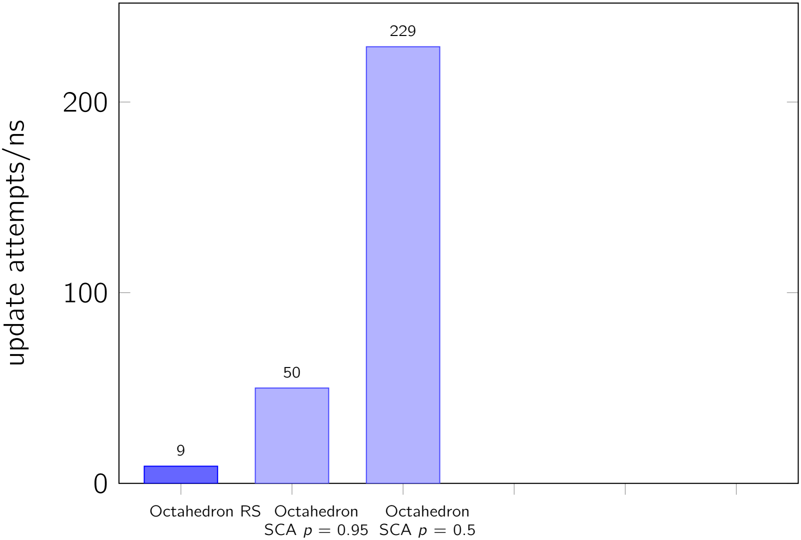
Random Sequential (RS)
on GPU: domain decomposition



+ uncorrelated updates

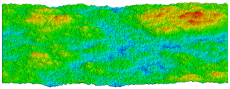
Kelling, J., Ódor, G., Gemming, S. *Comp. Phys. Commun.* **220** 205 (2017)
Kelling, J., Ódor, G., et al. *EPJST* **89** 175 (2012)

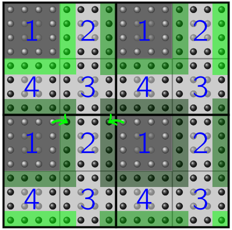
Performance



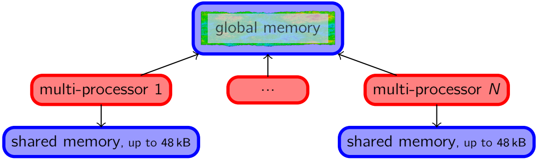
GPU: GTX Titan Black

Bit-Coded KPZ on GPUs

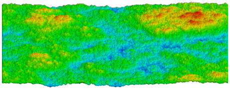


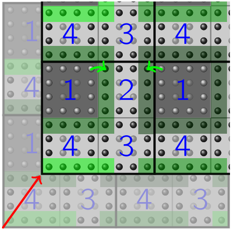


- double-tiling

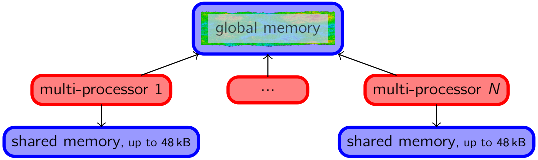


Bit-Coded KPZ on GPUs

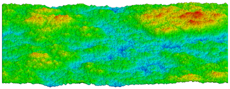


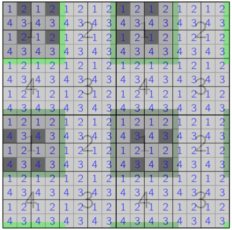


- double-tiling
  ... with random origin

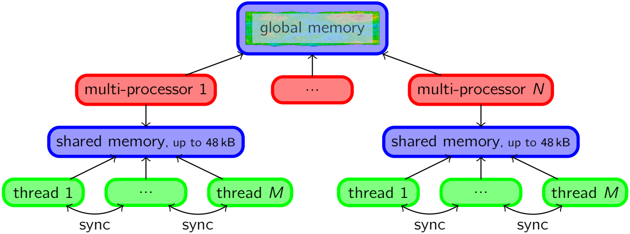


Bit-Coded KPZ on GPUs





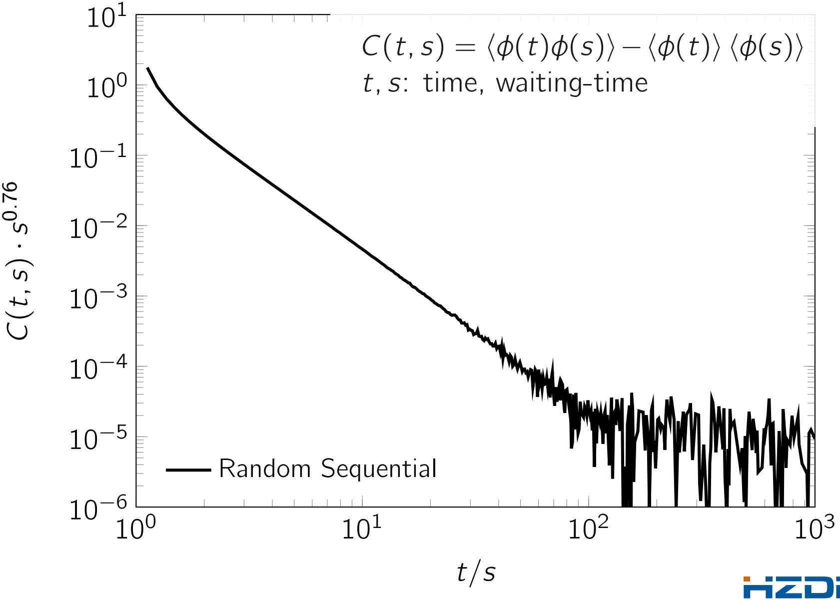
- double-tiling
  … with random origin



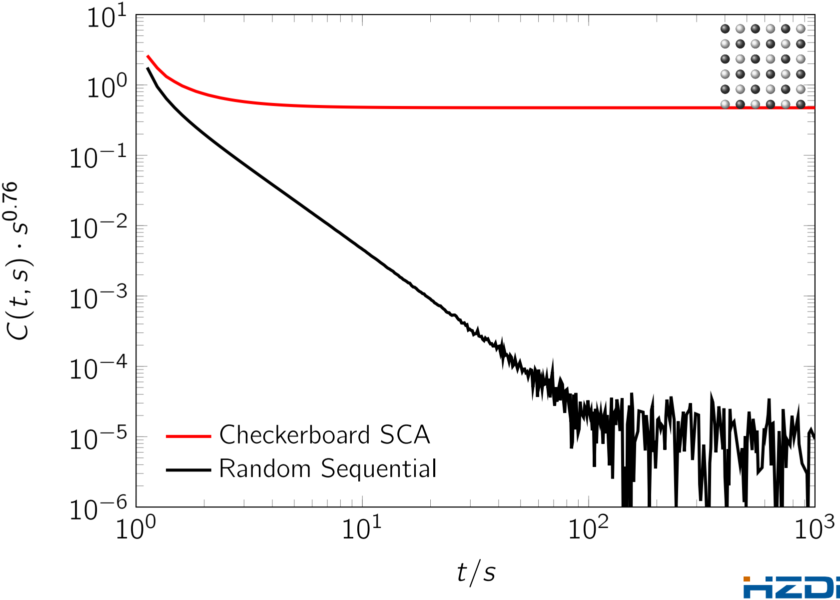
Parallel random sequential updates are hard.

Why should we care for them?

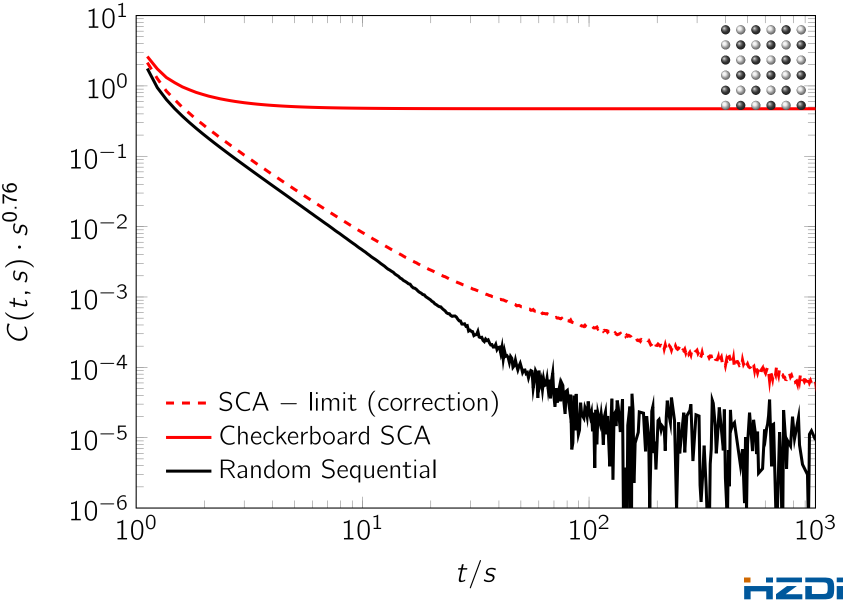
Auto-Correlation of a Lattice Gas



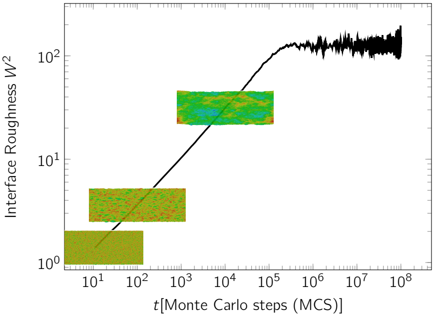
Auto-Correlation of a Lattice Gas



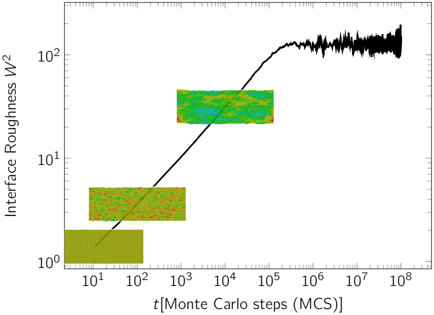
Auto-Correlation of a Lattice Gas

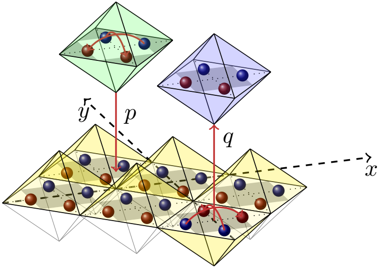


KPZ–Equation for Surface Growth



KPZ–Equation for Surface Growth



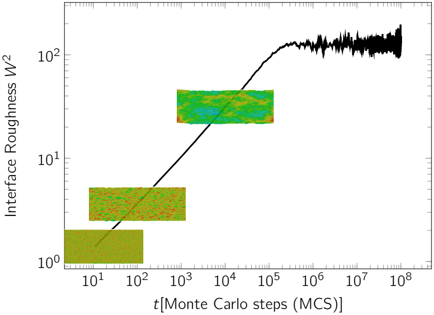


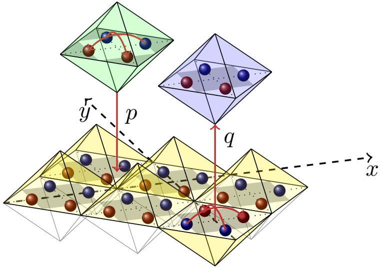
2 + 1D octahedron model

Ódor, G., Liedke, B., Heinig, K.-H. *Phys. Rev. E* **79** 021125 (2009)

HZDR

KPZ–Equation for Surface Growth





2 + 1D octahedron model

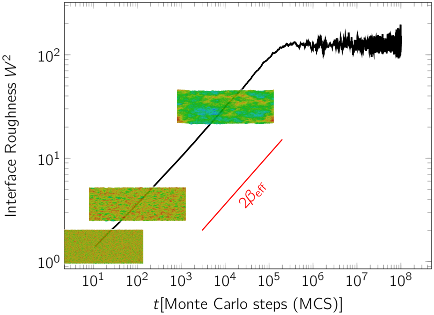
Ódor, G., Liedke, B., Heinig, K.-H. *Phys. Rev. E* **79** 021125 (2009)

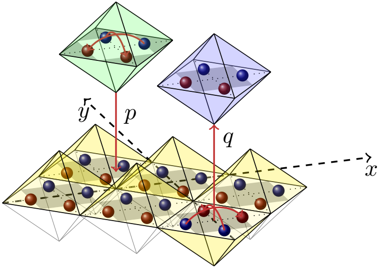
$$\mathrm{d}_t h(\mathbf{x},t) = \underbrace{v}_{\text{mean growth vel.}} + \underbrace{\sigma_2 \nabla^2 h(\mathbf{x},t)}_{\text{surface tension}} + \underbrace{\lambda\,[\nabla h(\mathbf{x},t)]^2}_{\text{local growth vel.}} + \underbrace{\eta(\mathbf{x},t)}_{\text{noise}}$$

Kardar–Parisi–Zhang stochastic differential equation

Kardar, M., Parisi, G., Zhang, Y.-C. *Phys. Rev. Lett.* **56** 889 (1986)

KPZ–Equation for Surface Growth





2 + 1D octahedron model

Ódor, G., Liedke, B., Heinig, K.-H. *Phys. Rev. E* **79** 021125 (2009)

$$\mathrm{d}_t h(\mathbf{x},t) = \underbrace{v}_{\text{mean growth vel.}} + \underbrace{\sigma_2 \nabla^2 h(\mathbf{x},t)}_{\text{surface tension}} + \underbrace{\lambda\,[\nabla h(\mathbf{x},t)]^2}_{\text{local growth vel.}} + \underbrace{\eta(\mathbf{x},t)}_{\text{noise}}$$

Kardar–Parisi–Zhang stochastic differential equation

Kardar, M., Parisi, G., Zhang, Y.-C. *Phys. Rev. Lett.* **56** 889 (1986)

β and the Kim–Kosterlitz Hypothesis

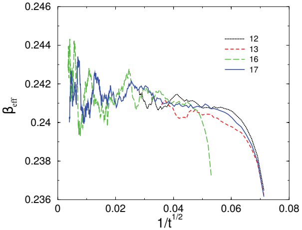
$$\beta = 1/4?$$

Kim, J. M., Kosterlitz, J. M. *Phys. Rev. Lett.* **62** 2289 (1989)

octahedron model

$$\Delta h = \pm 1$$

$$\beta < 1/4$$



Kelling, J., Ódor, G. *Phys. Rev. E* **84** 061150 (2011)

β and the Kim–Kosterlitz Hypothesis

$$\beta = 1/4?$$

Kim, J. M., Kosterlitz, J. M. *Phys. Rev. Lett.* **62** 2289 (1989)

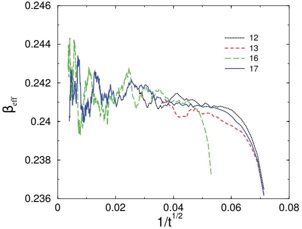
octahedron model

$$\Delta h = \pm 1$$

restricted solid-on-solid model

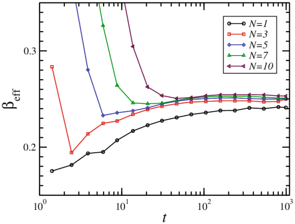
$$\Delta h \leq N$$

$$\beta < 1/4$$



Kelling, J., Ódor, G. *Phys. Rev. E* **84** 061150 (2011)

$$\beta \approx 1/4 \text{ for } N > 1?$$



Kim, J. M. *J. Korean Phys. Soc.* **67**(9) 1529 (2015)

β and the Kim–Kosterlitz Hypothesis

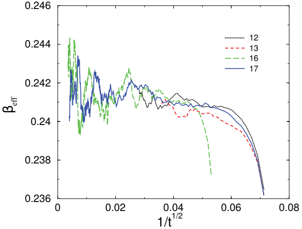
$$\beta = 1/4?$$

Kim, J. M., Kosterlitz, J. M. *Phys. Rev. Lett.* **62** 2289 (1989)

octahedron model

$$\Delta h = \pm 1$$

$$\beta < 1/4$$

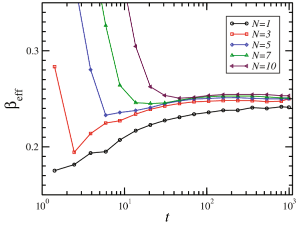


Kelling, J., Ódor, G. *Phys. Rev. E* **84** 061150 (2011)

restricted solid-on-solid model

$$\Delta h \leq N$$

$$\beta \approx 1/4 \text{ for } N > 1?$$



Kim, J. M. *J. Korean Phys. Soc.* **67**(9) 1529 (2015)

We need more states.

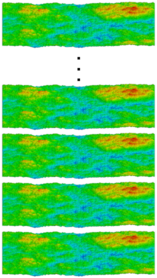
Part 2

Trivial parallism vs. SIMT

Handling more states.

Trivial parallism vs. SIMT

- efficient simulation of independent copies



Trivially parallel

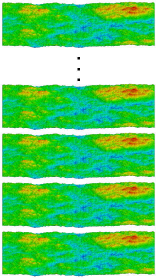
$\mapsto$ large samples $\Rightarrow$ good statistics

$\mapsto$ large parameter studies

$\mapsto$ large sets of initial conditions

Trivial parallism vs. SIMT

- efficient simulation of independent copies



Trivially parallel

$\mapsto$ large samples $\Rightarrow$ good statistics

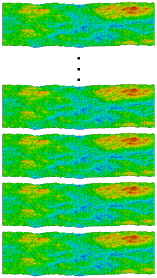
$\mapsto$ large parameter studies

$\mapsto$ large sets of initial conditions

$+$ random site-selection

Trivial parallism vs. SIMT

- efficient simulation of independent copies



Trivially parallel $\rightarrow$ **Multi-Surface**

$\mapsto$ large samples $\Rightarrow$ good statistics

$\mapsto$ large parameter studies

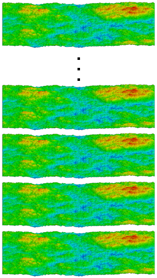
$\mapsto$ large sets of initial conditions

$+$ random site-selection

Ito, N., Kanada, Y. *Supercomputer* **3**(25) 1988

Trivial parallism vs. SIMT

- efficient simulation of independent copies



vector of $32, \ldots, 128, 256, \ldots$ layers
depending on application

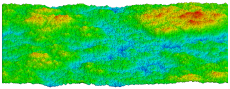
$\Rightarrow$ “random” accesses to vectors in global memory

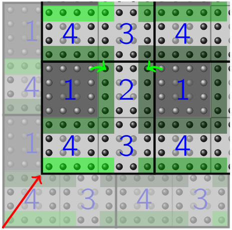
$\Rightarrow$ no caching of simulation state required

$\Rightarrow$ very efficient use of GPUs

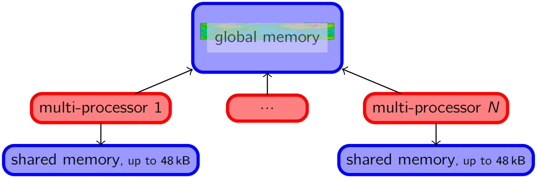
(vector processors/data parallelism)

Multi-Surface Approach for GPUs

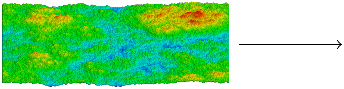


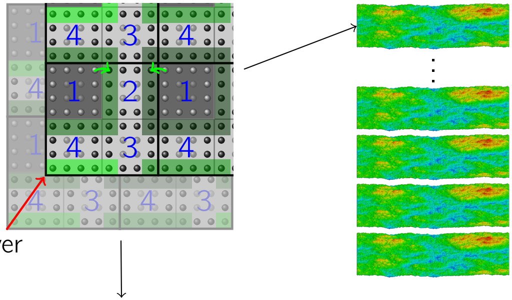


- double-tiling at device layer
  ... with random origin

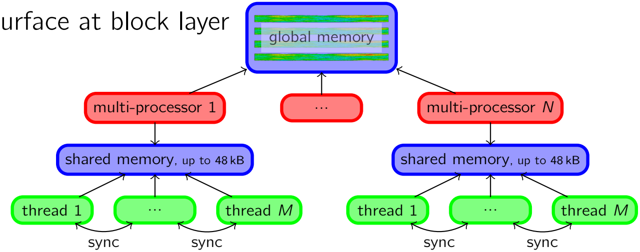


Multi-Surface Approach for GPUs



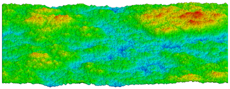


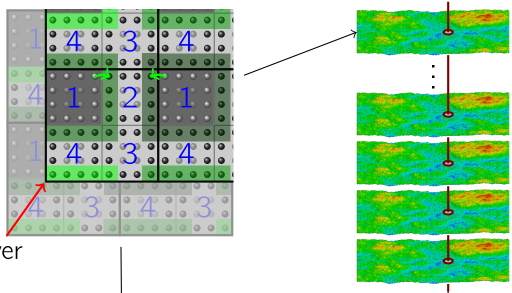
- double-tiling at device layer
  … with random origin
- Multi-Surface at block layer



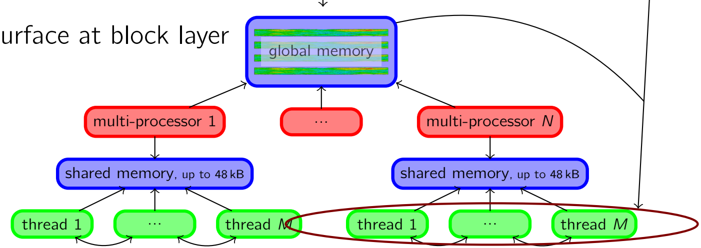
Jeffrey Kelling, Géza Ódor, Martin Weigel, Sibylle Gemming | FWIO | http//www.hzdr.de

Multi-Surface Approach for GPUs





- double-tiling at device layer
  ... with random origin
- Multi-Surface at block layer



HZDR

Decorrelating Samples

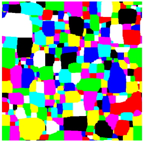
- random site-selection is about introducing uncorrelated noise
- we want to average over **independent** samples

Decorrelating Samples

- random site-selection is about introducing uncorrelated noise
- we want to average over **independent** samples
- domain growth, phase ordering: structure evolution

- random initial conditions
- independent random update acceptance

  (Boltzmann factors $\exp \Delta E / k_B T$)
- (quenched disorder)

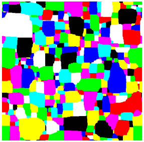


$\Rightarrow$ no problem

Jeffrey Kelling, Géza Ódor, Martin Weigel, Sibylle Gemming | FWIO | http//www.hzdr.de

Decorrelating Samples

- random site-selection is about introducing uncorrelated noise
- we want to average over **independent** samples
- domain growth, phase ordering: structure evolution
  - random initial conditions
  - independent random update acceptance
    (Boltzmann factors $\exp \Delta E / k_B T$)
  - (quenched disorder)

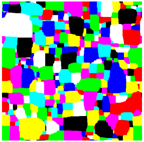


$\Rightarrow$ no problem

- surface growth
  - flat initial conditions
  - $\Rightarrow$ all simulations with identical site-selection would be identical

Decorrelating Samples

- random site-selection is about introducing uncorrelated noise
- we want to average over **independent** samples
- domain growth, phase ordering: structure evolution
  - random initial conditions
  - independent random update acceptance
    (Boltzmann factors $\exp \Delta E / k_B T$)
  - (quenched disorder)



$\Rightarrow$ no problem

- surface growth
  - flat initial conditions
  - $\Rightarrow$ all simulations with identical site-selection would be identical
  - randomly discard every 2nd update

Not Decorrelating Samples

Cases where identical noise across samples is desirable:

- sampling initial conditions
- calculating response functions

Not Decorrelating Samples

Cases where identical noise across samples is desirable:

- sampling initial conditions
- calculating response functions

* parallel annealing

RSOS Multi-Surface

- 8 bits per lattice-site are enough

$\Rightarrow$ process 4 packed samples per thread
4 bits per height-difference

RSOS Multi-Surface

- 8 bits per lattice-site are enough

⇒ process 4 packed samples per thread
4 bits per height-difference

word 0≡thread 0 word 1≡thread 1

...

RSOS Multi-Surface

- 8 bits per lattice-site are enough

$\Rightarrow$ process 4 packed samples per thread
4 bits per height-difference

$$\overbrace{\overbrace{(x,y)}^{\text{sample 0}}\overbrace{(x,y)}^{\text{sample 1}}\overbrace{(x,y)}^{\text{sample 2}}\overbrace{(x,y)}^{\text{sample 3}}}^{\text{word 0}\equiv\text{thread 0}}\;\overbrace{\overbrace{(x,y)}^{\text{sample 4}}\overbrace{(x,y)}^{\text{sample 5}}\overbrace{(x,y)}^{\text{sample 6}}\overbrace{(x,y)}^{\text{sample 7}}}^{\text{word 1}\equiv\text{thread 1}}\quad\ldots$$

Jeffrey Kelling, Géza Ódor, Martin Weigel, Sibylle Gemming | FWIO | http//www.hzdr.de

RSOS Multi-Surface

- 8 bits per lattice-site are enough

$\Rightarrow$ process 4 packed samples per thread
4 bits per height-difference

$$\overbrace{\underbrace{(x,y)}_{\text{sample 0}}\underbrace{(x,y)}_{\text{sample 1}}\underbrace{(x,y)}_{\text{sample 2}}\underbrace{(x,y)}_{\text{sample 3}}}^{\text{word 0}\equiv\text{thread 0}}\;\overbrace{\underbrace{(x,y)}_{\text{sample 4}}\underbrace{(x,y)}_{\text{sample 5}}\underbrace{(x,y)}_{\text{sample 6}}\underbrace{(x,y)}_{\text{sample 7}}}^{\text{word 1}\equiv\text{thread 1}}\quad\ldots$$

- randomly select 2 out of 4 samples for each thread

$\Rightarrow$ no idle threads

Collective Generation of Random Coordinates

- all threads access the same coordinate for each update

$\Rightarrow$ pre-compute list of update coordinates in shared memory

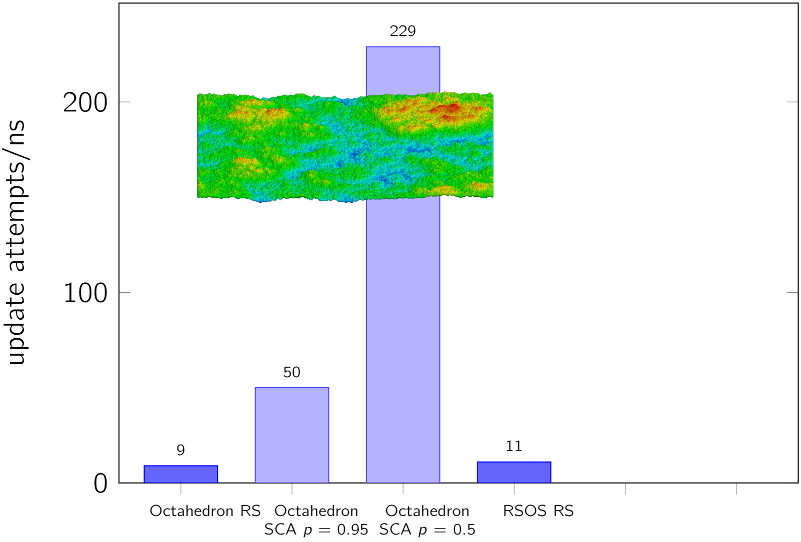
Collective Generation of Random Coordinates

- all threads access the same coordinate for each update

⇒ pre-compute list of update coordinates in shared memory

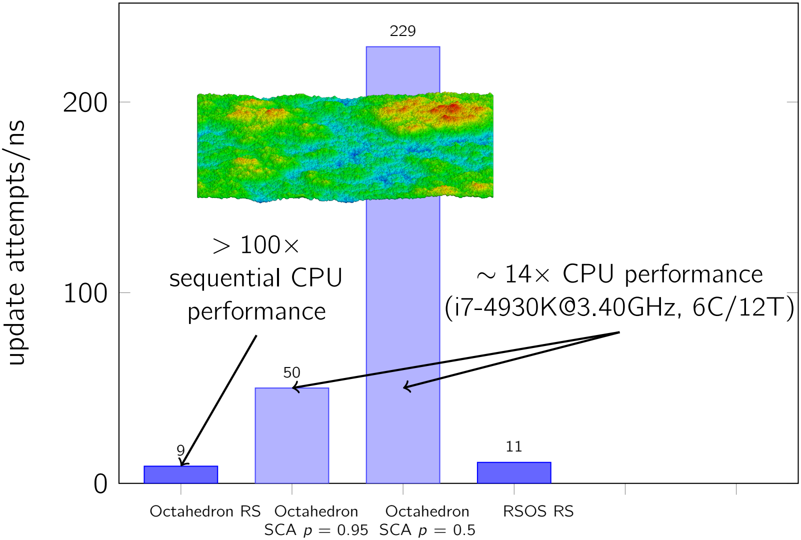
- each thread computes one component:
  1. generate random number
  2. apply transformations (origin shift, periodic boundary conditions)
- collectively refill list when used up

Performance



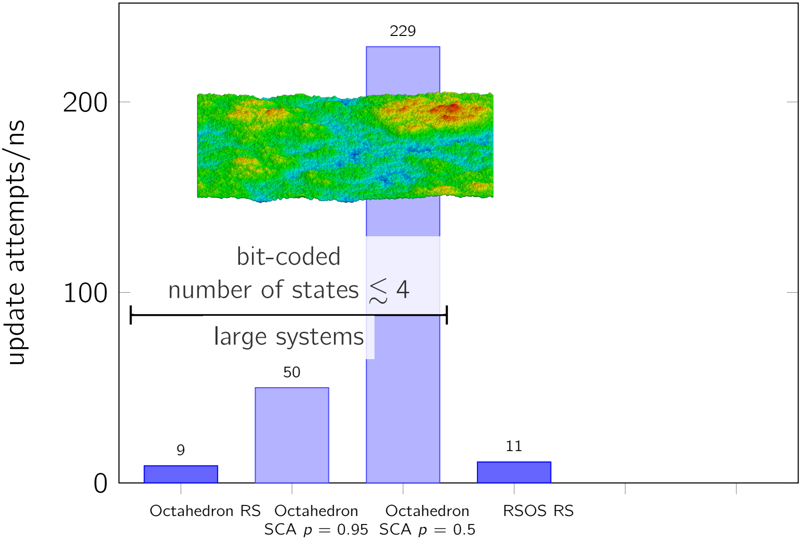
GPU: GTX Titan Black

Performance



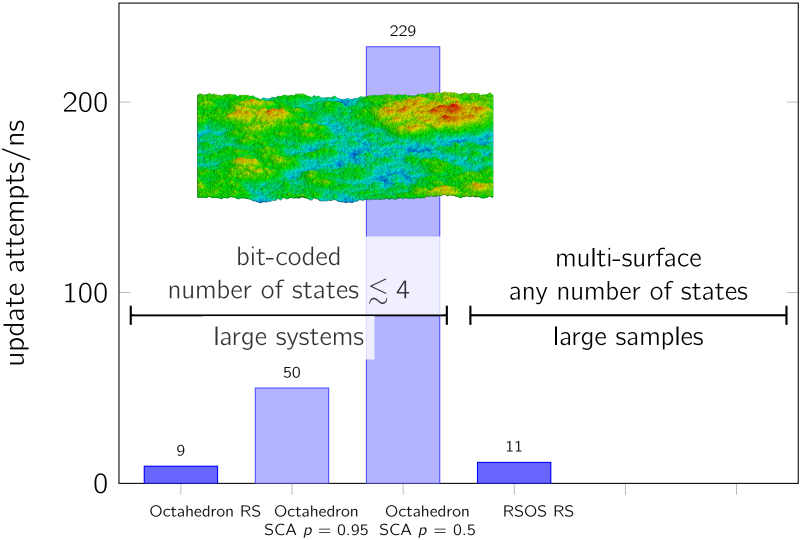
GPU: GTX Titan Black

Performance



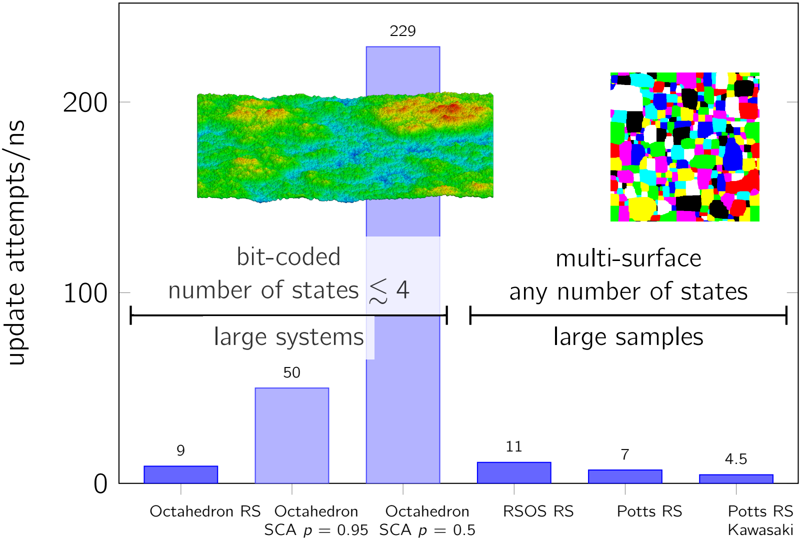
GPU: GTX Titan Black

Performance



GPU: GTX Titan Black

Performance



GPU: GTX Titan Black

Memory Limits: RSOS

- single-GPU implementations
- 64 threads per block

$\Rightarrow$ 256 samples

$\Rightarrow$ 256 B / MS lattice site

$\Rightarrow$ $2^{12} \times 2^{12}$ sites need 4 GB of gmem
$+$ random number generator states

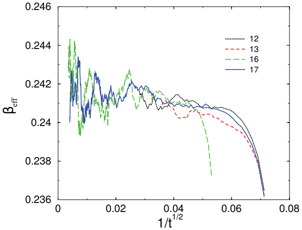
Memory Limits: Beyond

- consider: $2^{16} \times 2^{16}$ lattices sites, 2 bits per

$\Rightarrow$ 1 GB per sample

- efficient code would run > 1024 samples

akin to SCA: Kelling, J., Ódor, G., Gemming, S. *INES '16, IEEE* (2016)



Kelling, J., Ódor, G.
Phys. Rev. E **84** 061150 (2011)

Memory Limits: Beyond

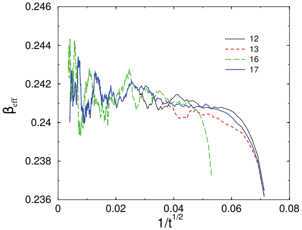
- consider: $2^{16} \times 2^{16}$ lattices sites, 2 bits per

$\Rightarrow$ 1 GB per sample

- efficient code would run > 1024 samples

  akin to SCA: Kelling, J., Ódor, G., Gemming, S. *INES '16, IEEE* (2016)

- our work actually needs this many samples



Kelling, J., Ódor, G.
Phys. Rev. E **84** 061150 (2011)

Memory Limits: Beyond

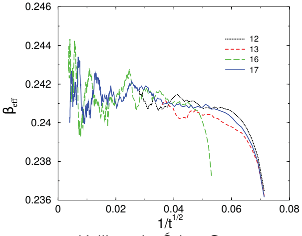
- consider: $2^{16} \times 2^{16}$ lattices sites, 2 bits per

$\Rightarrow$ 1 GB per sample

- efficient code would run > 1024 samples

akin to SCA: Kelling, J., Ódor, G., Gemming, S. *INES '16, IEEE* (2016)

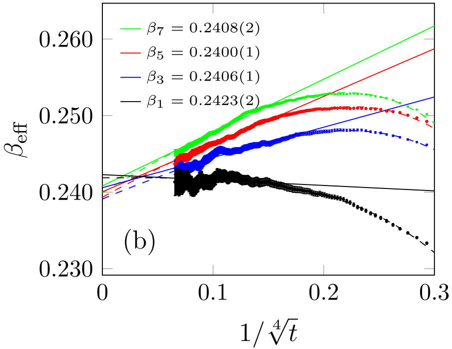
- our work actually needs this many samples
- spreading lattice across multiple GPUs more efficent then trival multi-GPU use



Kelling, J., Ódor, G.
Phys. Rev. E **84** 061150 (2011)

How did the β-thing turn out?

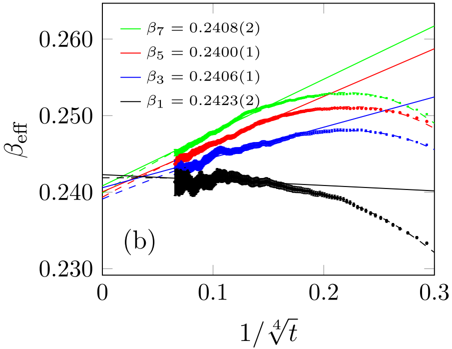
How did the β-thing turn out?



Kelling, J., Ódor, G., Gemming, S. *Phys. Rev. E* **94** 022107 (2016)

How did the β-thing turn out?

$\beta = 0.241(1)$ for all N



Kelling, J., Ódor, G., Gemming, S. *Phys. Rev. E* **94** 022107 (2016)

Conclusions and Outlook

- Think about vectorizing your trivial parallelism.

Conclusions and Outlook

- Think about vectorizing your trivial parallelism.
- we are developing a framework for Nd applications

Conclusions and Outlook

- Think about vectorizing your trivial parallelism.
- we are developing a framework for *N*d applications
- multi-GPU in the making
- … code will be made available after restructuring

Acknowledgements

- Artur Erbe
- Jörg Schuster
- Peter Zahn
- Henrik Schulz
- Nils Schmeißer
- Michael Bussmann
- Guido Juckeland

- my other colleagues
- computing time at ZIH Dresden, NIIF Hungary, HZDR Computing Center

NVIDIA GPU CENTER OF EXCELLENCE

J.Kelling@HZDR.de

Thank You.

This work has received funding from the Erasmus+ program via the Leonardo-Büro Sachsen and Coventry University.

Selected Publications

- Kelling, J., Ódor, G.:
  Extremely large-scale simulation of a Kardar-Parisi-Zhang model using graphics cards
  Phys. Rev. E **84** 061150 (2011)
- Kelling, J., Ódor, G., Nagy, M. F., Schulz, H., Heinig, K.-H.:
  Comparison of different parallel implementations of the 2+1-dimensional KPZ model and the 3-dimensional KMC model
  Eur. Phys. J. ST **210** 175 (2012)
- Kelling, J., Ódor, G., Gemming, S.:
  Bit-Vectorized GPU Implementation of a Stochastic Cellular Automaton Model for Surface Growth
  IEEE International Conference on Intelligent Engineering Systems (2016)
- Kelling, J., Ódor, G., Gemming, S.:
  Universality of 2+1 dimensional RSOS models
  Phys. Rev. E **94** 022107 (2016)
- Kelling, J., Ódor, G., Gemming, S.:
  Local scale-invariance of the $2 + 1$ dimensional Kardar–Parisi–Zhang model
  J. Phys. A **50** 12LT01 (2017)
- Kelling, J., Ódor, G., Gemming, S.:
  Suppressing correlations in massively parallel simulations of lattice models
  Comp. Phys. Commun. **220** 205 (2017)
- Kelling, J., Ódor, G., Gemming, S.:
  Dynamical universality classes of simple growth and lattice gas models
  J. Phys. A **51**(3) 035003 (2018)