Parallel Simulation of Apoptotic Receptor-Clustering on GPGPU Many-Core Architectures

Claus Braun*, Markus Daub†, Alexander Schöll*, Guido Schneider† and Hans-Joachim Wunderlich*

*Institute of Computer Architecture and Computer Engineering, University of Stuttgart,
Pfaffenwaldring 47, D-70569, Stuttgart, Germany
Email: {braun, wu}@informatik.uni-stuttgart.de

†Institute of Analysis, Dynamics, and Modeling, University of Stuttgart,
Pfaffenwaldring 57, D-70569, Stuttgart, Germany
Email: {Markus.Daub, Guido.Schneider}@mathematik.uni-stuttgart.de

***Abstract*—Apoptosis, the programmed cell death, is a physiological process that handles the removal of unwanted or damaged cells in living organisms. The process itself is initiated by signaling through *tumor necrosis factor* (TNF) receptors and ligands, which form clusters on the cell membrane. The exact function of this process is not yet fully understood and currently subject of basic research. Different mathematical models have been developed to describe and simulate the apoptotic receptor-clustering.**

In this interdisciplinary work, a previously introduced model of the apoptotic receptor-clustering has been extended by a new receptor type to allow a more precise description and simulation of the signaling process. Due to the high computational requirements of the model, an efficient algorithmic mapping to a modern many-core GPGPU architecture has been developed. Such architectures enable high-performance computing (HPC) simulation tasks on the desktop at low costs. The developed mapping reduces average simulation times from months to days (peak speedup of 256x), allowing the productive use of the model in research.

***Keywords*-GPGPU, parallel particle simulation, numerical modeling, apoptosis, receptor-clustering**

I. Introduction

Apoptosis, the programmed cell death, is an important physiological process since it enables organisms to remove unwanted or damaged cells. A deeper understanding of the processes involved in apoptosis is required for the control of cell death especially the initiation of the apoptotic signaling pathways. One of these pathways is the extrinsic pro-apoptotic signaling pathway which is initiated by signaling competent clusters of TNF receptors and the corresponding TNF ligands. In order to study the formation of signal competent clusters concerning their size and structure, a mathematical model for the motion and clustering of the receptors and ligands was introduced in [1]. Based on biological research results published in [2], the developed model with its receptor monomers and ligands has now been extended by a third particle type, the *receptor homodimers*. The original model has been developed and evaluated using MATLAB. A more efficient, grid-based algorithm has been implemented using the C programming language. However, being a stochastic model, a significant number of simulation runs is required to draw reliable conclusions, which causes very long computation times.

In recent years, traditional graphics processing units (GPUs) left the niche of visual data processing and emerged to highly parallel and fully programmable *many-core processor architectures*. The acronym *GPGPU* (General-Purpose Computations on Graphics Processing Units) has been coined for the use of such GPUs in compute-intensive, non-graphical applications. These many-core architectures are widely available at very low cost and they enable high-performance computing simulation applications on the desktop. Compared to classic CPU architectures, GPGPUs are optimized for high computational throughput and they gain their impressive performance through massive on-chip parallelism. A successful utilization of the high computational potential requires thorough adaption and optimization of the target algorithm to the GPGPU's hardware characteristics.

In this work, an efficient, parallel mapping of the introduced mathematical model to a GPGPU many-core architecture has been developed. It exploits fine-grained intra-GPU parallelism with multiple active simulation instances per GPGPU device, as well as coarse-grained inter-GPU parallelism by utilizing all available GPGPU devices within a system. The outcome of this interdisciplinary collaboration leads to a peak speedup of 256x and an overall speedup, which reduces simulation times from months to days and hours.

This paper is structured as follows: section II presents the related work in the field of apoptosis-related mathematical modeling and accompanying parallel simulations. Section III gives a motivation of the biological relevance of the extrinsic signaling pathway, followed by the description of the developed mathematical model in sections IV and V. The characteristics of modern GPGPU many-core architectures and the algorithmic mapping of the simulation model to such an architecture are presented in sections VI and VII. The evaluation of the mapping with detailed experimental results is given in section VIII.

978-1-4673-2560-8/12/$31.00 ©2012 IEEE

II. Related Work

Over the last years, several mathematical models of receptor-clustering on the cell surface have been established. In [3], the mathematical description of the cluster formation kinetics modeled by mass action kinetics for the receptor and ligand concentrations results in a system of coupled nonlinear differential equations. This ODE model only contains a temporal dependance and neglects the spatial extension of a single cell. Another model of the cluster formation introduced in [4] deals with thermodynamical aspects. In this model, the spatially extended cell membrane was modeled as a lattice and Monte Carlo simulations show the formation of receptor clusters. The idea of a model considering the spatial extension of the cell membrane was taken up in the lattice free particle model introduced in [1]. Additionally, the orientation of the particles is incorporated to the particle model. According to the biological research results published in [2], an extension of this particle model by a third particle type, the receptor dimers, is required.

A wide variety of approaches exists for particle and N-body simulations [5], which differ significantly in the way the interactions between the particles are computed and the particle data is organized. The basic principles range from *particle-particle/particle-mesh* approaches, *nested-grids*, *tree codes* to *self-consistent field methods* [6]. The fields of application with respect to GPGPUs include interactive and realtime visualizations of large particles sets [7], astrophysical applications, as well as numerous applications, e.g. in cellular modeling [8] and molecular dynamics [9]. However, to the best of our knowledge, there exists no accurate mathematical model for the description of apoptotic receptor-clustering, which has been mapped to GPGPU architectures.

III. Relevance of the Signaling Pathway

The central and executing machinery of apoptosis is a network of caspases, a specific form of proteases. Proteases are proteins being able to cleave and thereby activate other proteins. The extrinsic pro-apoptotic signaling pathway is one of two major signaling pathways which converge at the level of the caspase network, see Fig. 1. The external stimulus of the extrinsic signaling pathway is the activation of pro-caspase 8 at the signal competent ligand-receptor clusters on the cellular membrane. The ligand under consideration is TNF and the receptor belongs to the TNF receptor superfamily, e.g. TNF receptor of type 1 (TNFR1). According to [2], a so-called TNFR1-Fas chimera which consists of the extracellular domains of TNFR1 and the cytoplasmic part of the Fas receptor [10] is considered. The membrane distal cysteine rich domain of TNFR1-Fas receptors enables the multimerization of TNFR1-Fas receptors [11]. TNF exists as a homotrimer being able to bind up to three TNF receptors. The majority of the TNFR1-Fas receptors exist as pre-assembled homodimers and the minimal signal competent unit is one dimeric TNFR1-Fas receptor bound with two TNF ligands [2].

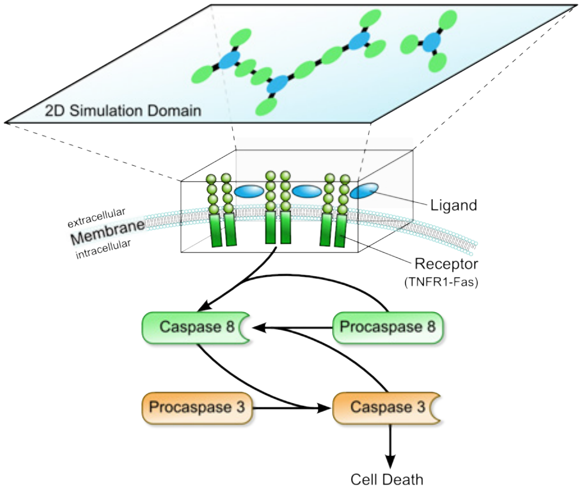

Figure 1: Extrinsic signaling pathway. Figure taken from [1].

The TNFR1-Fas receptors and the TNF ligands move around randomly on the cellular membrane. In order to investigate the formation of ligand-receptor clusters consisting of several signal competent units, a particle model describing the translatory and rotational motion of the molecules and appropriate binding conditions are required.

IV. Modeling of Receptor-Clustering

In the following, a particle model of the receptor-clustering with three different particle types is introduced. The basic idea of the particle model regarding monomeric TNFR1-Fas receptors and TNF ligands was established in [1]. Due to the suggestion of the existence of preformed dimeric TNFR1-Fas receptors [2], an extension of the model introduced in [1] by a third particle type, the dimeric TNFR1-Fas receptors, is desirable. Therefore, we present a particle model with three particles types, see Fig. 2.

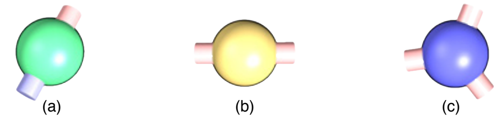

Figure 2: Different particle types - monomer (a), dimer (b) and ligand (c) - involved in the particle model.

For simplicity, the molecules are shortly called monomers, dimers and ligands. Starting point of the particle model are *Langevin* equations for the particle translation and the particle rotation

$$m\dot{\mathbf{v}} = \mathbf{F}_{\text{int}} - \beta\mathbf{v} + \sigma_{\text{trans}}\tilde{\mathbf{X}}_t,$$
$$I\dot{\omega} = g(\varphi, \mathbf{x}) - \gamma_{\text{rot}}\omega + \sigma_{\text{rot}}\tilde{D}_t,$$

where $\tilde{\mathbf{X}}_t$ and $\tilde{D}_t$ are white noises, the terms $-\beta\mathbf{v}$ and $-\gamma_{\text{rot}}\omega$ are friction terms and $\mathbf{F}_{\text{int}}$ and $g(\varphi, \mathbf{x})$ contain the interaction between the particles, cf. [1]. Here, the quantities $\mathbf{x}$ and φ with $\dot{\mathbf{x}} = \mathbf{v}$ and $\dot{\varphi} = \omega$ describe the motion of the particles. The dimensionless coordinates of the center of mass of the three particle types are denoted with

$\bar{\mathbf{x}}_{M/D/L}$ and the dimensionless angles for the description of the binding sites are $\bar{\varphi}_{M/D/L}$ summarized in the matrix $\bar{\boldsymbol{\xi}}$ and the vector $\bar{\boldsymbol{\varphi}}$, respectively. According to [1], neglecting the terms $m\dot{\mathbf{v}}$ and $I\dot{\omega}$, a system of stochastic differential equations with $3(z_{\mathrm{M}} + z_{\mathrm{D}} + z_{\mathrm{L}})$ equations can be derived

$$\begin{aligned} \mathrm{d}\bar{\mathbf{x}}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j} &= 3\tilde{\mu}^2 \bar{\mathbf{F}}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}})\mathrm{d}\bar{t} + \tilde{\mu}\mathrm{d}\widetilde{\mathbf{W}}_{\mathrm{tr},\bar{t},i/k/j}, \\ \mathrm{d}\bar{\varphi}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j} &= \frac{\kappa}{2}\tilde{\zeta}^2 g_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}})\mathrm{d}\bar{t} + \tilde{\zeta}\mathrm{d}\widetilde{W}_{\mathrm{r},\bar{t},i/k/j}, \end{aligned}$$

$1 \leq i \leq z_{\mathrm{M}}, 1 \leq k \leq z_{\mathrm{D}}, 1 \leq j \leq z_{\mathrm{L}}$, and $\tilde{\mu} = \sqrt{2}\mu$, $\tilde{\zeta} = \sqrt{2}\mu\zeta$. Here, $z_{\mathrm{M/D/L}}$ denotes the number of monomers, dimers and ligands, respectively. The stochastic dynamical behavior of the system is described by the Wiener processes $\widetilde{\mathbf{W}}_{\mathrm{tr},i/j/k}$ and $\widetilde{W}_{\mathrm{r},i/j/k}$, cf. [12], and the parameters μ, ζ, κ were derived in [1].

The interaction force $\bar{\mathbf{F}}_{\mathrm{int}}$ in dimensionless form is given by the Lennard-Jones-$(2n, n)$ potential $\bar{V}_{\mathrm{LJ}}(\bar{r}) = 2^{2n}\left(\bar{\sigma}_{\mathrm{LJ}}/\bar{r}\right)^{2n} - 2^{1-n}\left(\bar{\sigma}_{\mathrm{LJ}}/\bar{r}\right)^{n}$ [13] for the repulsive and attractive interaction between two monomers, a monomer and a ligand, and between a dimer and a ligand.

For the interaction between two dimers, two ligands and between a monomer and a dimer, there is only a repulsive interaction described by $\bar{W}_{\mathrm{LJ}}(\bar{r}) = 2^{2n}(\bar{\sigma}_{\mathrm{LJ}}/\bar{r})^{2n}$. Then, the force acting on a monomer is given by

$$\begin{aligned} \bar{\mathbf{F}}_{M_i}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}}) = &- \sum_{k=1}^{z_{\mathrm{D}}} \bar{W}'_{\mathrm{LJ}}(|\bar{\mathbf{x}}_{\mathrm{D}_k} - \bar{\mathbf{x}}_{\mathrm{M}_i}|)\mathbf{e}_{\mathrm{D}_k\mathrm{M}_i} \\ &- \sum_{k=1, k\neq i}^{z_{\mathrm{M}}} \big(\bar{V}'_{\mathrm{LJ}}(|\bar{\mathbf{x}}_{\mathrm{M}_k} - \bar{\mathbf{x}}_{\mathrm{M}_i}|) \cdot H(\delta - |\psi_{\mathrm{M}_k;\,\mathrm{M}_i}|) \\ &+ \bar{W}'_{\mathrm{LJ}}(|\bar{\mathbf{x}}_{\mathrm{M}_k} - \bar{\mathbf{x}}_{\mathrm{M}_i}|) \cdot H(|\psi_{\mathrm{M}_k;\,\mathrm{M}_i}| - \delta)\big)\, \mathbf{e}_{\mathrm{M}_k\mathrm{M}_i} \\ &- \sum_{k=1}^{z_{\mathrm{L}}} \big(\bar{V}'_{\mathrm{LJ}}(|\bar{\mathbf{x}}_{\mathrm{L}_k} - \bar{\mathbf{x}}_{\mathrm{M}_i}|) \cdot H(\delta - |\psi_{\mathrm{L}_k;\,\mathrm{M}_i}|) \\ &+ \bar{W}'_{\mathrm{LJ}}(|\bar{\mathbf{x}}_{\mathrm{L}_k} - \bar{\mathbf{x}}_{\mathrm{M}_i}|) \cdot H(|\psi_{\mathrm{L}_k;\,\mathrm{M}_i}| - \delta)\big)\, \mathbf{e}_{\mathrm{L}_k\mathrm{M}_i}, \end{aligned} \tag{1}$$

$i = 1, \ldots, z_{\mathrm{M}}$, with $\psi_{\mathrm{M}_k;\,\mathrm{M}_i} := \bar{\varphi}_{\mathbf{e}_{\mathrm{M}_k\mathrm{M}_i}} - \bar{\varphi}_{\mathrm{M}_k;\,\mathrm{M}_i}$ and $\psi_{\mathrm{L}_k;\,\mathrm{M}_i} := \bar{\varphi}_{\mathbf{e}_{\mathrm{L}_k\mathrm{M}_i}} - \bar{\varphi}_{\mathrm{L}_k;\,\mathrm{M}_i}$. Here, $\bar{\varphi}_{\mathbf{e}_{\mathrm{M}_k\mathrm{M}_i}}$ denotes the angle between the vector $\mathbf{e}_{\mathrm{M}_k\mathrm{M}_i}$ and the positive real line and $\bar{\varphi}_{\mathrm{M}_k;\,\mathrm{M}_i}$ describes the angle between the compatible binding site of M_k lying the closest to $\mathbf{e}_{\mathrm{M}_k\mathrm{M}_i}$ and the positive real line. Furthermore, $H(\cdot)$ denotes the Heaviside-function and δ is the size of the apex angle of the binding sites. The force acting on a dimer and a ligand has a similar structure as (1) according to the repulsive and attractive interaction explained above.

Finally, the function g that ensures the correct relative orientation of the particles has to be defined precisely for the different particle types. Let $R_{\mathrm{M}_i;\,\mathrm{M}_k} = |\mathbf{e}_{\mathrm{M}_i\mathrm{M}_k}|$ and $R_{\mathrm{M}_i;\,\mathrm{L}_k} = |\mathbf{e}_{\mathrm{M}_i\mathrm{L}_k}|$. Then, the function g for monomers is given by

$$\begin{aligned} g_{M_i}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}}) = &\frac{1}{z_{\mathrm{M}}} \sum_{\substack{k=1 \\ k\neq i}}^{z_{\mathrm{M}}} \psi_{\mathrm{M}_i;\,\mathrm{M}_k}(\psi_{\mathrm{M}_i;\,\mathrm{M}_k} - \pi)(\psi_{\mathrm{M}_i;\,\mathrm{M}_k} + \pi) \cdot \\ &H(r_{\mathrm{cut}} - R_{\mathrm{M}_i;\,\mathrm{M}_k}) \frac{r_{\mathrm{cut}} - R_{\mathrm{M}_i;\,\mathrm{M}_k}}{r_{\mathrm{cut}}} \cdot \frac{3\sqrt{3}}{\pi^3} \\ &+ \frac{1}{z_{\mathrm{L}}} \sum_{k=1}^{z_{\mathrm{L}}} \psi_{\mathrm{M}_i;\,\mathrm{L}_k}(\psi_{\mathrm{M}_i;\,\mathrm{L}_k} - \pi)(\psi_{\mathrm{M}_i;\,\mathrm{L}_k} + \pi) \cdot \\ &H(r_{\mathrm{cut}} - R_{\mathrm{M}_i;\,\mathrm{L}_k}) \frac{r_{\mathrm{cut}} - R_{\mathrm{M}_i;\,\mathrm{L}_k}}{r_{\mathrm{cut}}} \cdot \frac{3\sqrt{3}}{\pi^3}, \end{aligned}$$

where $r_{\mathrm{cut}} > 0$ is a suitably chosen constant which is explained below (sect.VII). The functions $g_{D_k}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}})$ and $g_{L_j}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}})$ for dimers and ligands have a similar structure. With the functions $\bar{\mathbf{F}}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}})$ and $g_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}})$, the interaction between the particles is completely described.

V. Euler-Maruyama Approximation of SDE

In order to simulate the receptor-clustering, the system of nonlinearly coupled systems of stochastic differential equations is solved numerically. Here, an *Euler-Maruyama* approximation of the stochastic differential equations, as described in [14], is used

$$\begin{aligned} \Delta\bar{\mathbf{x}}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j} &= 3\tilde{\mu}^2 \bar{\mathbf{F}}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}})\Delta\bar{t} + \tilde{\mu}\Delta\widetilde{\mathbf{W}}_{\mathrm{tr},\bar{t},i/k/j}, \\ \Delta\bar{\varphi}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j} &= \frac{\kappa}{2}\tilde{\zeta}^2 g_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}(\bar{\boldsymbol{\xi}}, \bar{\boldsymbol{\varphi}})\Delta\bar{t} + \tilde{\zeta}\Delta\widetilde{W}_{\mathrm{r},\bar{t},i/k/j}, \end{aligned}$$

$1 \leq i \leq z_M$, $1 \leq k \leq z_D$, $1 \leq j \leq z_L$. The terms $[\Delta\widetilde{\mathbf{W}}_{\mathrm{tr},\bar{t},i/j/k}]_l, l = 1, 2$, and $\Delta\widetilde{W}_{\mathrm{r},\bar{t},i/j/k}$ are normally distributed random variables with mean zero and variance $\tau\Delta t$, i.e. $\mathcal{N}(0, \tau\Delta t)$, and can be written as $\Delta\widetilde{W}_{\bar{t}} = Z_{\bar{t}} \cdot \sqrt{\tau\Delta t}$, where $Z_{\bar{t}} \sim \mathcal{N}(0, 1)$. Besides, the terms $\Delta\bar{\mathbf{x}}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}$ and $\Delta\bar{\varphi}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}$ describe the discrete variation in the particle coordinates and the variation in the orientation of the particle binding site, respectively.

Thus, the simulation of the particle translation and particle rotation requires the generation of $\mathcal{N}(0, 1)$ distributed random variables and the evaluation of the interaction forces $\bar{\mathbf{F}}_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}$ and the functions $g_{\mathrm{M}_i/\mathrm{D}_k/\mathrm{L}_j}$. An appropriate binding condition finalizes the simulation of the receptor-clustering.

VI. GPGPU Background

GPGPU architectures gain their impressive computational performance through massive on-chip parallelism and throughput-optimization. Large numbers of rather simple arithmetic processing cores are tightly coupled to form multiprocessor units. These multiprocessors provide small shared memories and L1-caches that enable fast communication and data re-use among the cores. They are able to communicate via L2-caches and the global memory of the GPGPU device. Modern GPGPU architectures combine coarse-grained scalable data and task parallelism at multiprocessor-level with fine-grained data and thread parallelism at core-level.

GPGPU programs are partitioned into *host code* that runs on the host's CPU and one or more *parallel kernels*, which run on the GPGPU. Fig. 3 shows the basic thread execution and memory organization by the example of Nvidia's CUDA (Compute Unified Device Architecture).

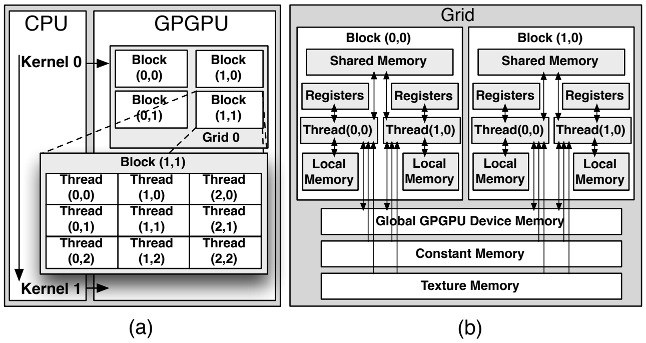

Figure 3: GPGPU thread and memory organization.

Kernels execute sequential programs with a set of very lightweight threads. Hence, the problem is mapped to a grid of *thread blocks*, which themselves consist of a grid of threads (see Fig. 3 (a)). A thread block is processed by a multiprocessor. Fig. 3 (b) shows the accompanying memory organization. Each thread has a private set of registers and a very fast local memory. All threads within a thread block have low latency access to the shared memory and the L1-cache of the multiprocessor. Besides the local memories, GPGPUs typically offer a global memory, as well as cached constant and texture memories, which can be accessed by all threads of the GPGPU.

VII. Algorithmic Mapping to GPGPU

The extended mathematical model has been mapped to a grid-based, stochastic particle simulation, which can consist of up to three different kinds of particles (monomers, dimers, ligands) in different numbers. The forces and torsional moments are computed in parallel for all particles. Thereby, each particle is processed by a single thread. The algorithmic mapping extends the scope of parallelism from pure thread-level to multiple parallel simulation instances on a single GPGPU and the utilization of multiple GPGPUs.

The simulation domain is subdivided into equally sized cells and all particles of the simulation reside within these cells. According to the model, the interactions between particles decrease with increasing distance. This allows the introduction of a circular *interaction region*, defined by a distance threshold value $r_{\text{cut}} > 0$. Particles with a distance $d < r_{\text{cut}}$ interact with each other, while particles with a distance $d > r_{\text{cut}}$ do not interact. This reduces the computational overhead since only interacting particles have to be processed.

Memory bandwidth is a scarce resource on GPGPUs and high data locality is required for efficient memory accesses. Therefore, the grid-based mapping stores all particles in sorted lists according to the grid cells they reside in. The sorting is performed in parallel on the GPGPU using radix-sort. Although the data locality is improved by the grid-based organization of the particles, the number of required memory accesses is still very high and associated with notable costs. The developed mapping introduces two main optimizations, which help to improve performance at this point.

Following a Brownian motion, the particles do not tend to leave the boundaries of a grid cell within each simulation time step, if the grid cell size is reasonably large. Therefore, the system checks each particle's position during the motion computation. If the particle leaves its current grid cell, a sorting flag is set. After the motion computation, only the marked particles have to be sorted. This *position-dependent sorting* reduces the overall number of sorting steps in each simulation time step.

Particles within a grid cell can interact not only with particles inside this cell, but with particles of up to nine neighboring cells. Without further optimization, the mapping would have to access the particle data from these nine grid cells, with one read direction at a time. This means that at first all neighboring cells in the upper left direction are read, followed by all cells in top direction, followed by the upper right direction, etc. In total, each grid cell is read nine times, which induces a very high memory bandwidth overhead. Therefore, the mapping re-schedules the memory accesses in a way that the particles of each grid cell are read only once. Fig. 4 shows the optimized scheduling for an example with 3 × 3 grid cells. At first, the upper left grid cell is read by all its direct neighbors, then the upper middle grid cell and so on. This scheduling utilizes the GPGPU's shared memories for data re-use, since the considered particle data can be accessed by all the neighboring grid cells after being loaded once.

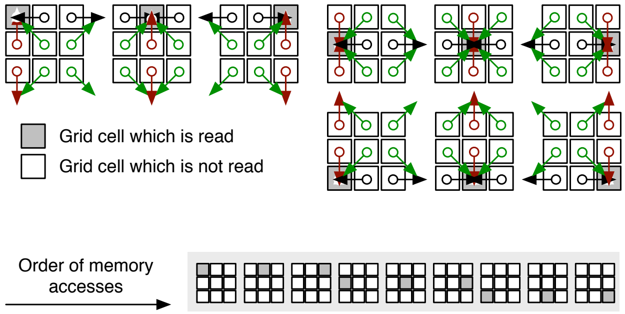

Figure 4: Optimized scheduling of read accesses to particle data.

A third optimization subdivides the grid cells into four quadrants to simplify the treatment of particles in neighboring grid cells. Fig. 5 shows that there exist exactly four possibilities for the position of a particle's interaction region within a grid cell, as long as $2 \cdot r_{\text{cut}} \leq grid\_cell\_size$. In this case, the particles within the quadrants interact with a maximum of three neighboring grid cells' particles and

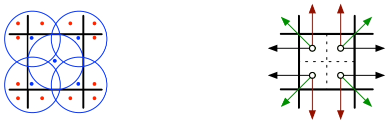

Figure 5: Subdivision of grid cells into four quadrants.

the particles of their own grid cell. This reduces the overall number of memory accesses from nine to four (see Fig. 6).

The size of the grid cells has a direct influence on the achievable simulation performance. Essentially, two aspects dominate the costs of each simulation step: the sorting algorithm and the computation of the interactions. Thus, the costs are describes by the formula $\mathcal{O}(3N) \cdot \mathcal{O}(NM) + \mathcal{O}(4\frac{N}{M^2})$, where N is the total number of particles and M denotes the number of grid cells. The first term in the formula indicates the costs for the sorting algorithm (radixsort) multiplied by the likeliness for one particle leaving a grid cell. Here, we assume that a duplication of M or N leads to a duplication of the likeliness under the assumption of a uniform particle distribution. The second summand in the formula describes the costs for the computation of the interactions which is performed in parallel. Each particle interacts with particles in four grid cells and we assume that each grid cell contains N/M^2 particles, hence, the term $4N/M^2$. It is easy to check that the function $f_N(M) = 3N \cdot \alpha NM + 4 \cdot N/M^2$ possesses a minimum which is reached for $M = 30$ with $\alpha = 0.05/(N \cdot M_{max})$ and $1/M_{max} \geq 2 \cdot r_{\text{cut}}$. Actually, the optimal number of grid cells fits very well to the observations of the computer experiments.

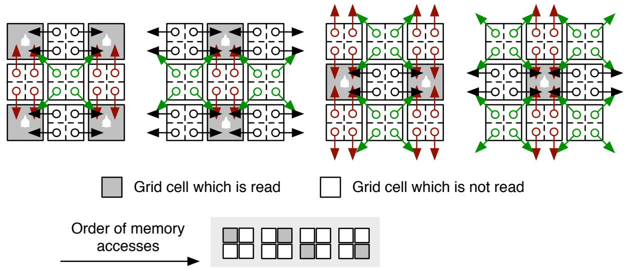

Figure 6: Reduced number of grid cell read accesses.

Fig. 7 gives an overview of the sequence of algorithmic steps, which is performed during a simulation. At first, the available GPGPUs are allocated and the required particle data is transferred. This step is performed only once at the beginning of the simulation. In the second step, the main simulation loop is entered with the generation of a set of normally distributed random numbers. These pseudorandom numbers are computed in parallel on the GPGPU. The random numbers are required for the later computation of the new particle positions. In the third algorithmic step, the forces and torsional moments are computed for all the particles in the system. Arithmetically, the formulas introduced in sections IV and V are evaluated. The fourth step comprises the computation of the new particle positions, using the generated random numbers and the computed forces. In the fifth and last step of the main simulation loop, the grid is updated and the lists are sorted to keep the data locality. The main simulation loop is repeated until the total number of demanded simulation runs has been computed. The dotted boxes *sampling* and *cluster detection* in Fig. 7 indicate that the simulation time steps are sampled at user-defined intervals. During such a sampling step, the particle bindings are evaluated to detect potential cluster structures.

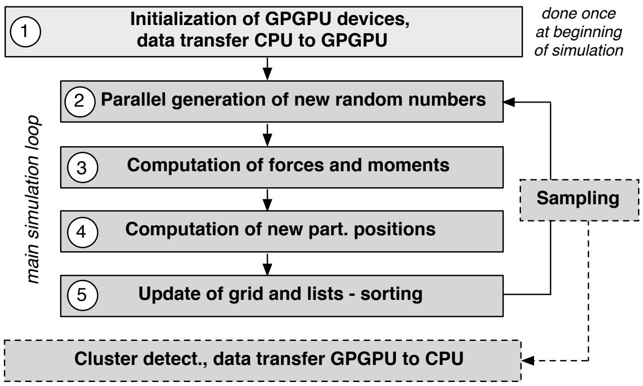
1
Initialization of GPGPU devices, data transfer CPU to GPGPU
done once at beginning of simulation
2
Parallel generation of new random numbers
3
Computation of forces and moments
4
Computation of new part. positions
5
Update of grid and lists - sorting
main simulation loop
Sampling
Cluster detect., data transfer GPGPU to CPU

Figure 7: Structure of the parallel simulation system.

VIII. Experimental Results

The developed mapping has been evaluated with respect to the achievable computational performance and the biological indications of the pro-apoptotic receptor-clustering. The used hardware platform consists of an Intel Xeon X5680 CPU (3.33GHz) and four Nvidia C2070 GPGPUs (1.15GHz), running Linux and CUDA 4.1. The receptor-clustering results and the particle-related figures in this paper have been visualized with *CellVis* [1].

A. Evaluation of Computational Performance

The overall achievable performance of the simulation model has been evaluated through multiple simulation runs over different numbers of particles. Each run consisted of 10^5 simulation time steps. Table I shows the speedups for a setup with the same number of monomer, dimer and ligand particles. The results show that the overall speedup increases significantly with larger numbers of particles. This result was expected and is mainly due to the fact, that each particle is processed by a single thread. As described in section VII, the computational potential of the GPGPU device is not only exploited through parallel calculations of forces, torsional moments and random numbers, but also by execution of multiple parallel simulation instances on a single GPGPU and the utilization of multiple GPGPU devices in a system. For this evaluation, a typical simulation setup with 2496 monomers, 2496 dimers and 1344 ligands has been chosen. Table II shows the achieved speedups for configurations with up to 8 parallel simulations per device and 4 GPGPU devices.

Total particles	9216	18432	36864	64512
CPU time (s)	8546	31412	103412	348904
GPGPU time (s)	182	325	685	1363
Speedup	**46.96x**	**96.65x**	**150.97x**	**255.98x**

Table I: Speedup over different numbers of particles.

The results show a slightly lower speedup compared to the single instance, single GPGPU setup, which is mainly due to the increased administrative overhead for the management of multiple simulation instances on multiple devices.

Par. Instances	4	8	16	24	32
CPU (s)	20920	41840	83680	125520	167360
GPGPU (s)	208	305	520	760	977
Speedup	**101.1x**	**131.98x**	**163.44x**	**163.65x**	**170.43x**

Table II: Speedup with different numbers of parallel instances on multiple GPGPUs.

B. *Evaluation of Biological Indication*

To analyze the formation of ligand-receptor clusters, a simulation with 192 monomers, 2688 dimers and 2880 ligands has been performed. The assumption that the majority of the receptors are pre-assembled as homodimers is taken into account and the number of particles under consideration is of a realistic biological order. r_{cut} is set to 10^{-3} and the apex angle of the sector modeling the binding site is chosen to be $\delta = \pi/3$. The exponent in the Lennard-Jones potential is $n = 6$ and the other parameter values are $\mu = 0.094, \zeta = 2887$ and $\kappa = 10^4$. For a simulation time of 0.1 s with a time step $\Delta\bar{t} = 10^{-9}$, the computation time on a single GPGPU took only 1.95 days. On this time scale of simulation time, the formation of ligand-receptor clusters is observed and signal competent cluster units consisting of one dimer bound with two ligands are formed. The cluster after 62.30 ms of size 6 consists of two signal competent cluster units, the cluster of size 4 contains one signal competent unit, see Figure 8. Besides the clusters of size 4 and size 6 illustrated in Figure 8, a non-negligible number of ligand-receptor clusters of size 3 occurs. However, not all of these ligand-receptor clusters of size 3 are signal competent. Since the formation of ligand-receptor clusters highly depends on the number of particles, a detailed analysis of the receptor clustering in dependence of the particle configuration is desirable and will be possible due to the mapping of the algorithms to the GPGPU architecture.

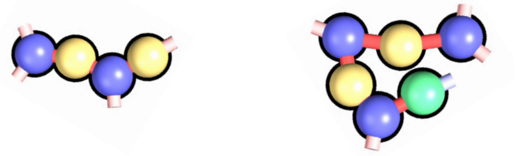

Figure 8: Ligand-receptor clusters of size 4 and size 6.

IX. Conclusion

In this paper, the joint effort of an interdisciplinary collaboration for the precise, parallel simulation of pro-apoptotic receptor-clustering has been presented. A previously introduced mathematical model has been extended and improved by a third particle type, the receptor homodimers. The high computational requirements of the model motivated the development of an algorithmic mapping to a modern GPGPU accelerator architecture. Besides a peak speedup of more than 256x, the parallelization leads to a significant reduction of computing times for most simulation setups of interest. Generally speaking, such configurations are now simulated in hours or days instead of months.

X. Acknowledgment

The authors would like to thank the German Research Foundation (DFG) for financial support of their projects within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

References

[1] M. Falk *et al.*, “Modeling and Visualization of Receptor Clustering on the Cellular Membrane”, in *2011 IEEE Symposium on Biological Data Visualization (BioVis)*, October 2011, pp. 9–15.

[2] V. Boschert *et al.*, “Single chain TNF derivatives with individually mutated receptor binding sites reveal differential stoichiometry of ligand receptor complex formation for TNFR1 and TNFR2”, *Cellular Signalling*, vol. 22, no. 7, pp. 1088–1096, 2010.

[3] A. S. Perelson and C. DeLisi, “Receptor clustering on a cell surface. I. Theory of receptor cross-linking by ligands bearing two chemically identical functional groups”, *Mathematical Biosciences*, vol. 48, no. 1-2, pp. 71–110, 1980.

[4] C. Guo and H. Levine, “A Thermodynamic Model for Receptor Clustering”, *Biophysical Journal*, vol. 77, no. 5, pp. 2358–2365, 1999.

[5] J. Barnes and P. Hut, “A hierarchical O(N log N) force-calculation algorithm”, *Nature*, vol. 324, no. 4, pp. 446–449, 1986.

[6] L. Hernquist and J. Ostriker, “A Self-Consistent Field Method for Galactic Dynamics”, *The Astrophysical Journal*, vol. 386, pp. 375–397, 1992.

[7] P. Kipfer, M. Segal, and R. Westermann, “UberFlow: a GPU-Based Particle Engine”, in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, ser. HWWS ’04. New York, NY, USA: ACM, 2004, pp. 115–122.

[8] P. Richmond *et al.*, “High Performance Cellular Level Agent-Based Simulation with FLAME for the GPU”, *Briefings in Bioinformatics*, vol. 11, no. 3, pp. 334–347, 2010.

[9] R. G. Belleman, J. Bedorf, and S. F. P. Zwart, “High Performance Direct Gravitational N-Body Simulations on Graphics Processing Units II: An Implementation in CUDA”, *New Astronomy*, vol. 13, no. 2, pp. 103 – 112, 2008.

[10] A. Krippner-Heidenreich *et al.*, “Control of Receptor-induced Signaling Complex Formation by the Kinetics of Ligand/Receptor Interaction”, *Journal of Biological Chemistry*, vol. 277, no. 46, pp. 44 155–44 163, 2002.

[11] M. Branschädel *et al.*, “Dual function of cysteine rich domain (crd) 1 of tnf receptor type 1: Conformational stabilization of crd2 and control of receptor responsiveness”, *Cellular signalling*, vol. 22, no. 3, pp. 404–414, 2010.

[12] C. Gardiner, *Handbook of stochastic methods: for physics, chemistry & the natural sciences,*, ser. Springer Series in Synergetics. Springer, 2004, vol. 13.

[13] J. E. Jones, “On the Determination of Molecular Fields. I. From the Variation of the Viscosity of a Gas with Temperature”, *Proc. of the Royal Society of London. Series A*, vol. 106, no. 738, pp. 441–462, 1924.

[14] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations (Stochastic Modelling and Applied Probability)*, corrected ed. Springer, August 1992.