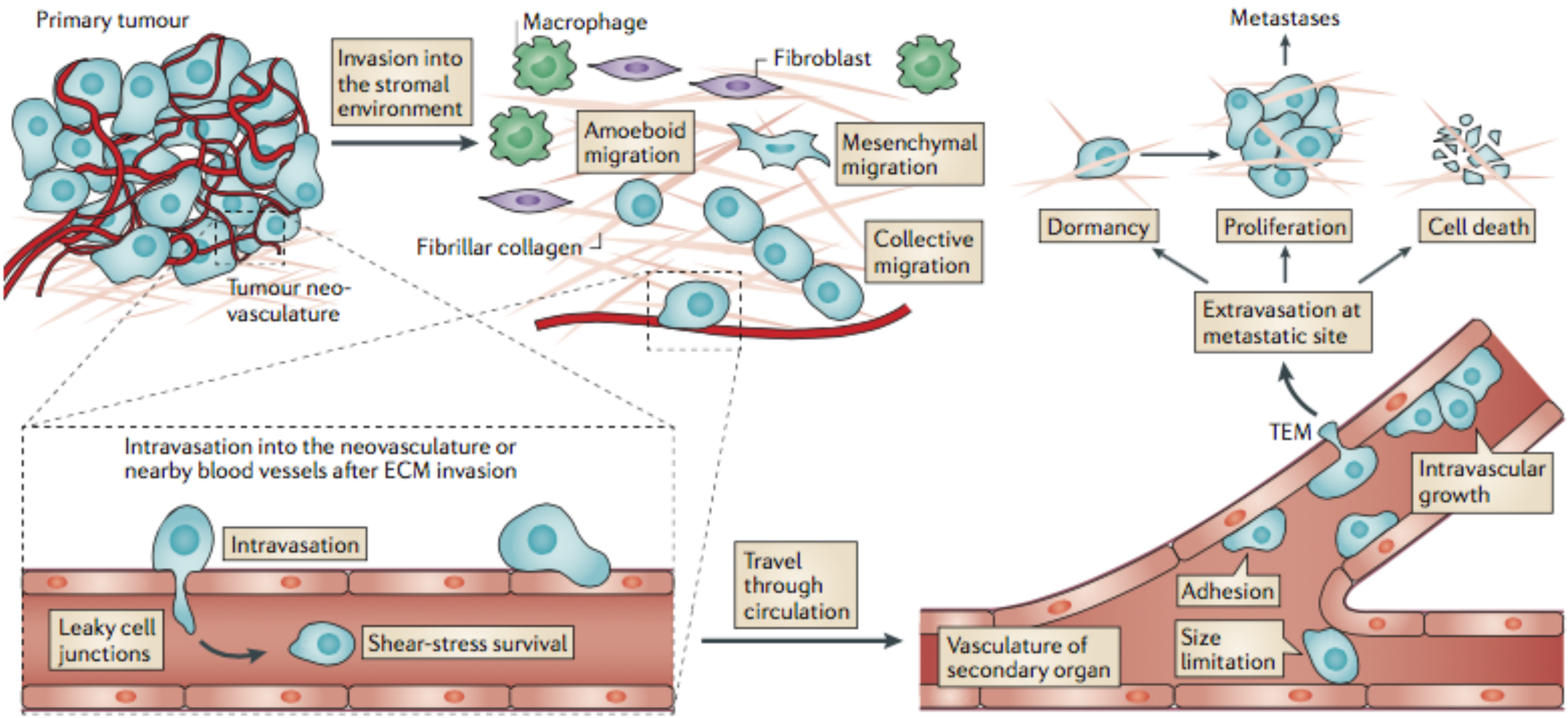
uDeviceX

18’688 K20Xs
RUNNING AFTER A TUMOR CELL

Diego Rossinelli, ETH Zurich

The In-Silico Lab-on-a-Chip: Petascale and High-Throughput Simulations of Microfluidics at Cell Resolution
Rossinelli, Tang, Lykov, Alexeev, Bernaschi, Hadjidoukas, Bisson, Joubert, Conti, Karniadakis, Fatica, Pivkin, Koumoutsakos
ACM Gordon Bell finalist 2015

METASTASIS AND CTCS



NATURE REVIEWS | CANCER

VOLUME 13 | DECEMBER 2013 | 859

- 8 million cancer deaths every year
- 90% attributed to metastasis

CTCs in the blood

-> higher risks of metastasis

How to detect the presence of CTCs in a blood sample?

THE CTC-ICHIP

Blood-Borne Metastasis

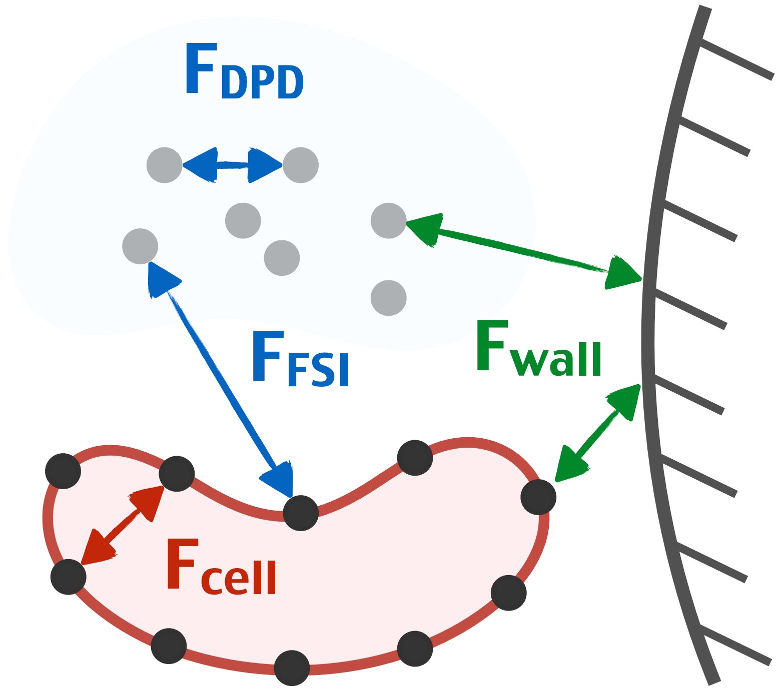
↓

Circulating Tumor Cells
(CTCs)

NUMERICAL METHOD

DISSIPATIVE PARTICLE DYNAMICS

$$\mathbf{F}_i = \sum_{n=1,n\neq i}^{N} \mathbf{F}_{i,n}^{C,\mathrm{DPD}} + \mathbf{F}_{i,n}^{D,\mathrm{DPD}} + \mathbf{F}_{i,n}^{R,\mathrm{DPD}}$$
$$+ \sum_{k=1,k\neq i}^{K} \mathbf{F}_{i,k}^{C,\mathrm{FSI}} + \mathbf{F}_{i,k}^{D,\mathrm{FSI}} + \mathbf{F}_{i,k}^{R,\mathrm{FSI}}$$
$$+ \sum_{m=1,m\neq i}^{M} \mathbf{F}_{i,m}^{C,\mathrm{wall}} + \mathbf{F}_{i,m}^{D,\mathrm{wall}} + \mathbf{F}_{i,m}^{R,\mathrm{wall}}$$

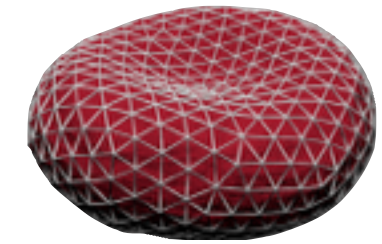


$$\mathbf{F}_{ij}^{C} = \begin{cases} a_{ij}(1-r_{ij})\mathbf{e}_{ij}, & \text{if } r_{ij} < 1 \\ 0, & \text{if } r_{ij} \geq 1 \end{cases}$$
$$\mathbf{F}_{ij}^{D} = -\gamma w^{D}(r_{ij})(\mathbf{e}_{ij}\cdot\mathbf{v}_{ij})\mathbf{e}_{ij}$$
$$\mathbf{F}_{ij}^{R} = \sigma w^{R}(r_{ij})\theta_{ij}\mathbf{e}_{ij}$$

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$$
$$r_{ij} = ||\mathbf{r}_{ij}||$$
$$\mathbf{e}_{ij} = \mathbf{r}_{ij}/||\mathbf{r}_{ij}||$$
$$\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$$

$$w^{D}(r) = (w^{R}(r))^2$$
$$\sigma^2 = 2\gamma k_B T$$

FORCES IN THE SYSTEM: RBC



$$\mathbf{F}^{\text{cell}} = \sum_{n=1}^{N} \boldsymbol{F}^{\text{dihedral},1}_{0,n-1,n,n+1} + \boldsymbol{F}^{\text{dihedral},2}_{0,n,N+n,n+1} + \boldsymbol{F}^{\text{triangle}}_{0,n,n+1} + \boldsymbol{F}^{\text{bond}}_{0,n}$$

$$= \sum_{n=1}^{N} \beta^{b}_{n,n+1} \left[\frac{\boldsymbol{\xi}_n \times \boldsymbol{a}_{n+1} + \boldsymbol{\xi}_{n+1} \times \boldsymbol{a}_n}{\xi_n \xi_{n+1}} - \cos\theta_{n,n+1} \left(\frac{\boldsymbol{\xi}_n \times \boldsymbol{a}_n}{\xi_n^2} + \frac{\boldsymbol{\xi}_{n+1} \times \boldsymbol{a}_{n+1}}{\xi_{n+1}^2} \right) \right]$$

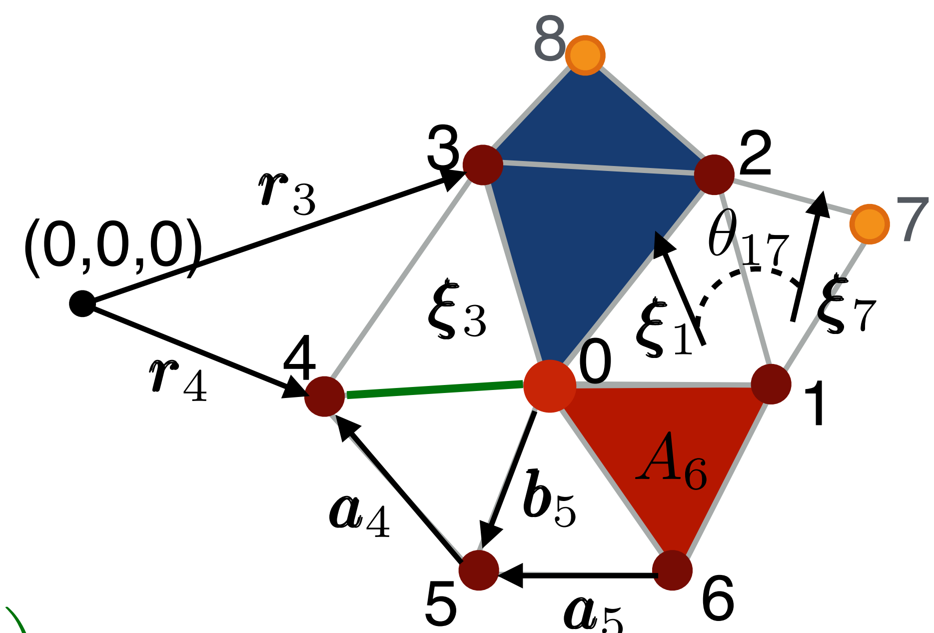
$$+ \beta^{b}_{n,N+n} \left[\frac{\boldsymbol{\xi}_{N+n} \times \boldsymbol{a}_n}{\xi_n \xi_{N+n}} - \cos\theta_{n,N+n} \frac{\boldsymbol{\xi}_n \times \boldsymbol{a}_n}{\xi_n^2} \right]$$

$$+ \left(\frac{qC_q}{A_n^{q+1}} - k_a \frac{A - A_0^{tot}}{A_0^{tot}} \right) \frac{\boldsymbol{\xi}_n \times \boldsymbol{a}_n}{4A_n}$$

$$- \frac{k_v}{18} \frac{V - V_0^{tot}}{V_0^{tot}} (\boldsymbol{\xi}_n + (\boldsymbol{r}_0 + \boldsymbol{r}_n + \boldsymbol{r}_{n+1}) \times \boldsymbol{a}_n)$$

$$- \frac{k_B T}{p} \left(\frac{1}{4(1 - b_n/l_m)^2} - \frac{1}{4} + \frac{b_n}{l_m} \right) \frac{\boldsymbol{b}_n}{b_n}$$

$$+ \sqrt{2k_B T} \left(\sqrt{2\gamma^T} d\overline{\boldsymbol{W}^S_{ij}} + \sqrt{3\gamma^C - \gamma^T} \frac{tr[d\boldsymbol{W}_{ij}]}{3} \boldsymbol{I} \right)$$



$$x_0 = l_0 / l_m$$

$$A_0 = \sqrt{3} l_0^2 / 4$$

$$C_q = \frac{\sqrt{3} A_0^{q+1} k_B T (4x_0^2 - 9x_0 + 6)}{4pql_m(1 - x_0)^2}$$

$$\beta^b_{ij} = k_b \frac{\sin\theta_{ij}\cos\theta_0 - \cos\theta_{ij}\sin\theta_0}{\sqrt{1 - \cos^2\theta_{ij}}}$$

$$d\overline{\boldsymbol{W}^S_{ij}} = d\boldsymbol{W}^S_{ij} - tr[d\boldsymbol{W}^S_{ij}]\mathbf{1}/3$$

ONE TIME STEP

Non-trivial communication

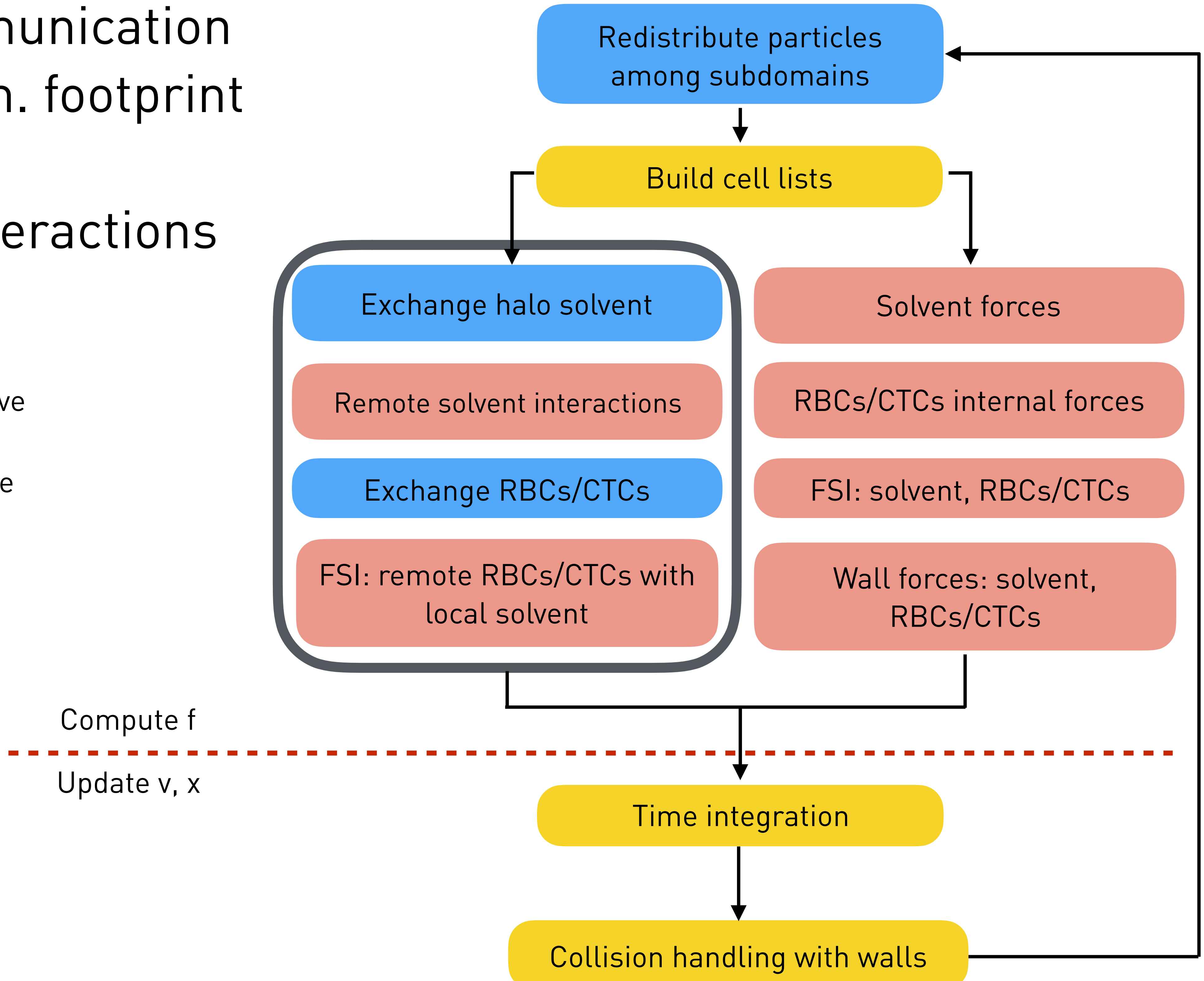
- 15-30MB comm. footprint
- 180 messages
- 5-40 million interactions

Compute intensive

Memory intensive

Communication

Remote forces

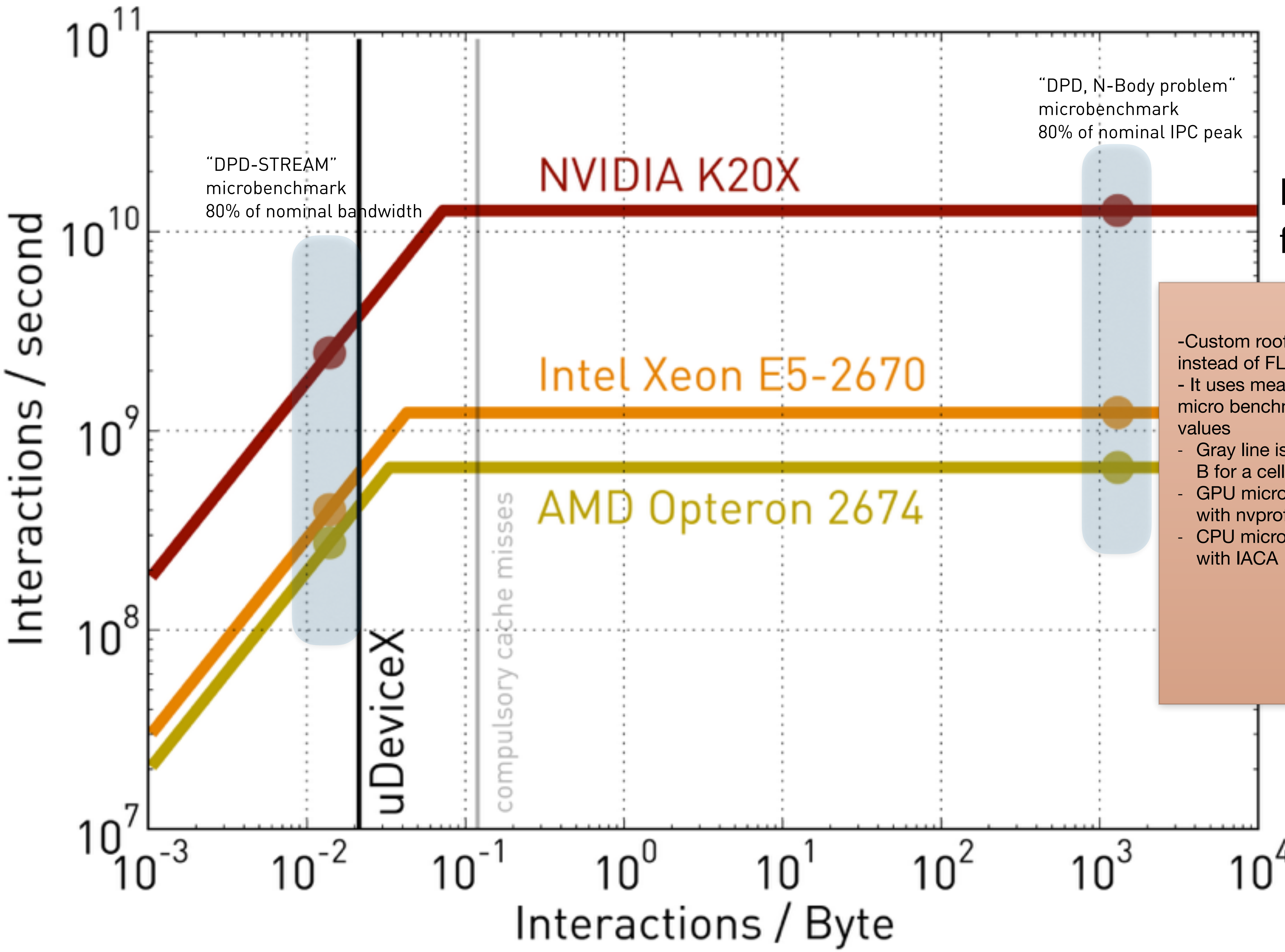


HPC

Minimize #interactions

Maximize interactions/s —> Maximize IPC

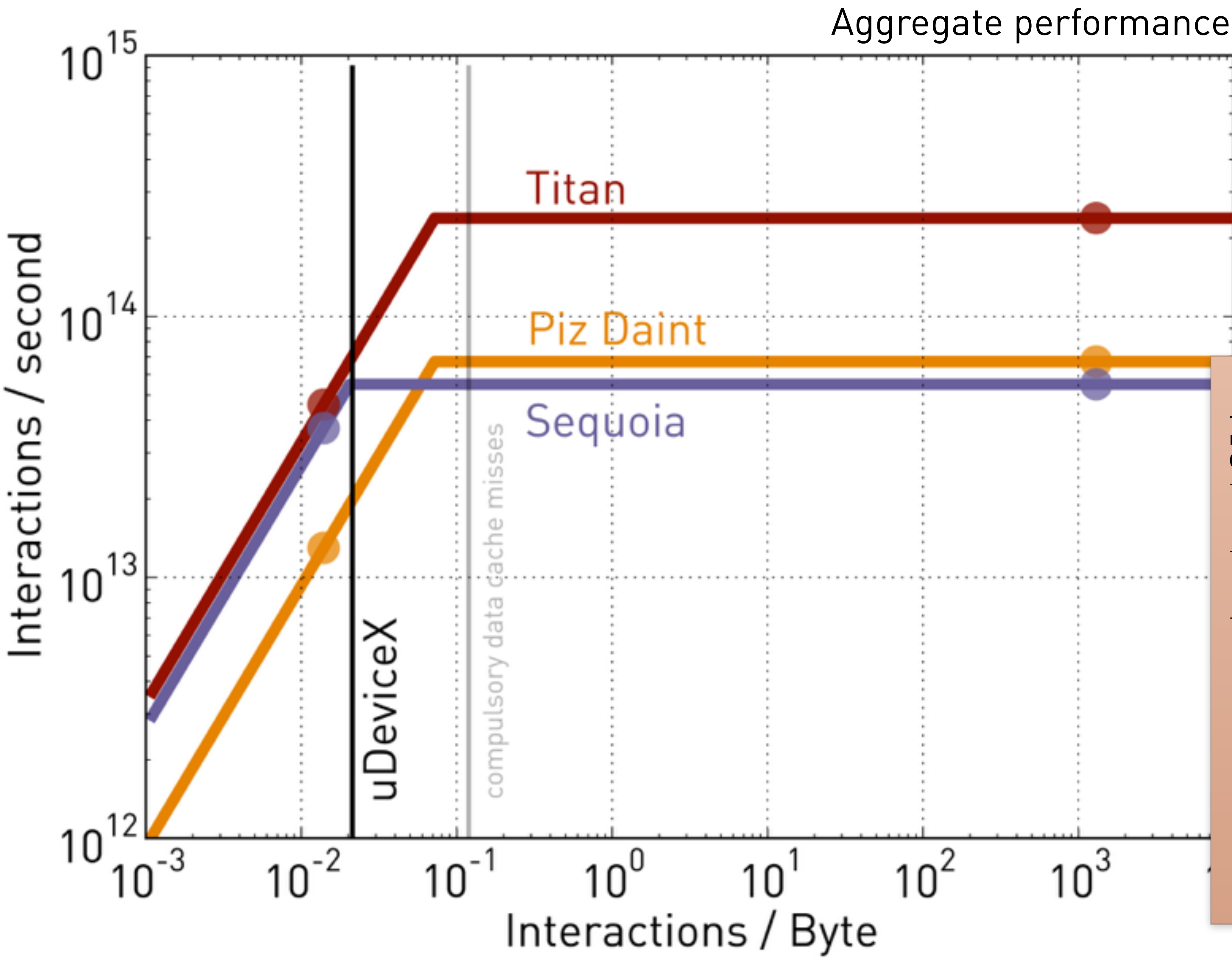
DPD: HARDWARE OF CHOICE



K20X:
fastest by 5-10X

- -Custom roofline: DPD interactions instead of FLOP
- - It uses measured values from the micro benchmarks and not the nominal values
  - Gray line is the upper bound of DPD/B for a cell-list based simulation
  - GPU microbenchmark were assessed with nvprof
  - CPU microbenchmark were assessed with IACA

DPD: SUPERCOMPUTER OF CHOICE



Best choice: Titan (less clear on the left)

- Although the difference with sequoia is not significant for simulations featuring 0.01 interactions per Byte
- Sequoia “links” Titan and Piz Daint: it has similar aggregate bandwidth as Titan and performance of Piz Daint
- Sequoia seems to be very profitable for real applications (often men-bound)
- remember that all this is about the UPPER BOUND, not the actual performance! K20X can still outperform BGQ!

see A28 and A29 of Patterson

SUPERCOMPUTING CHALLENGES

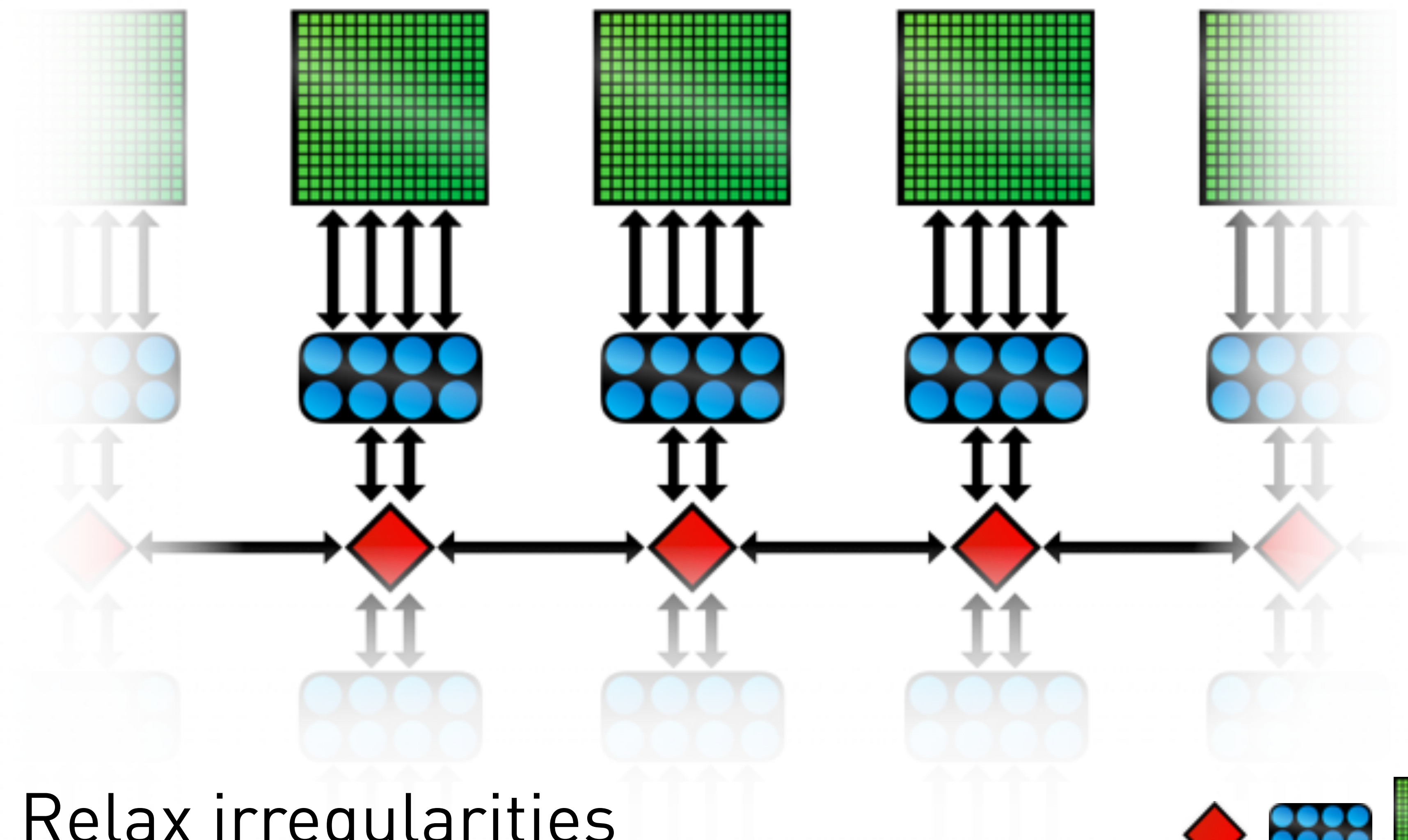
TTS = 1/freq x 1/IPC x #instr.

DPD	CHALLENGE
Rapidly changing neighbours	No gain from Verlet lists
Expensive forces	High instruction count
IOP-based random forces	Low IPC on the K20X
Irregular computation Irregular access patterns Irregular inter-rank messages	Warp divergence Uncoalesced access Penalised network performance Poor C/T overlap

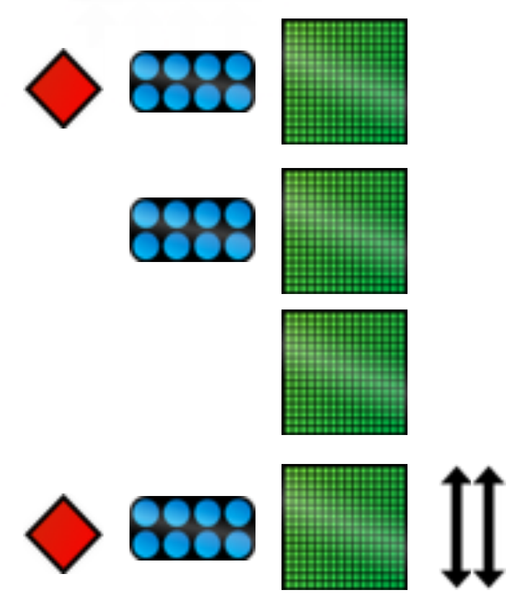
→ **Latency-bound performance**

→ **Low IPC**

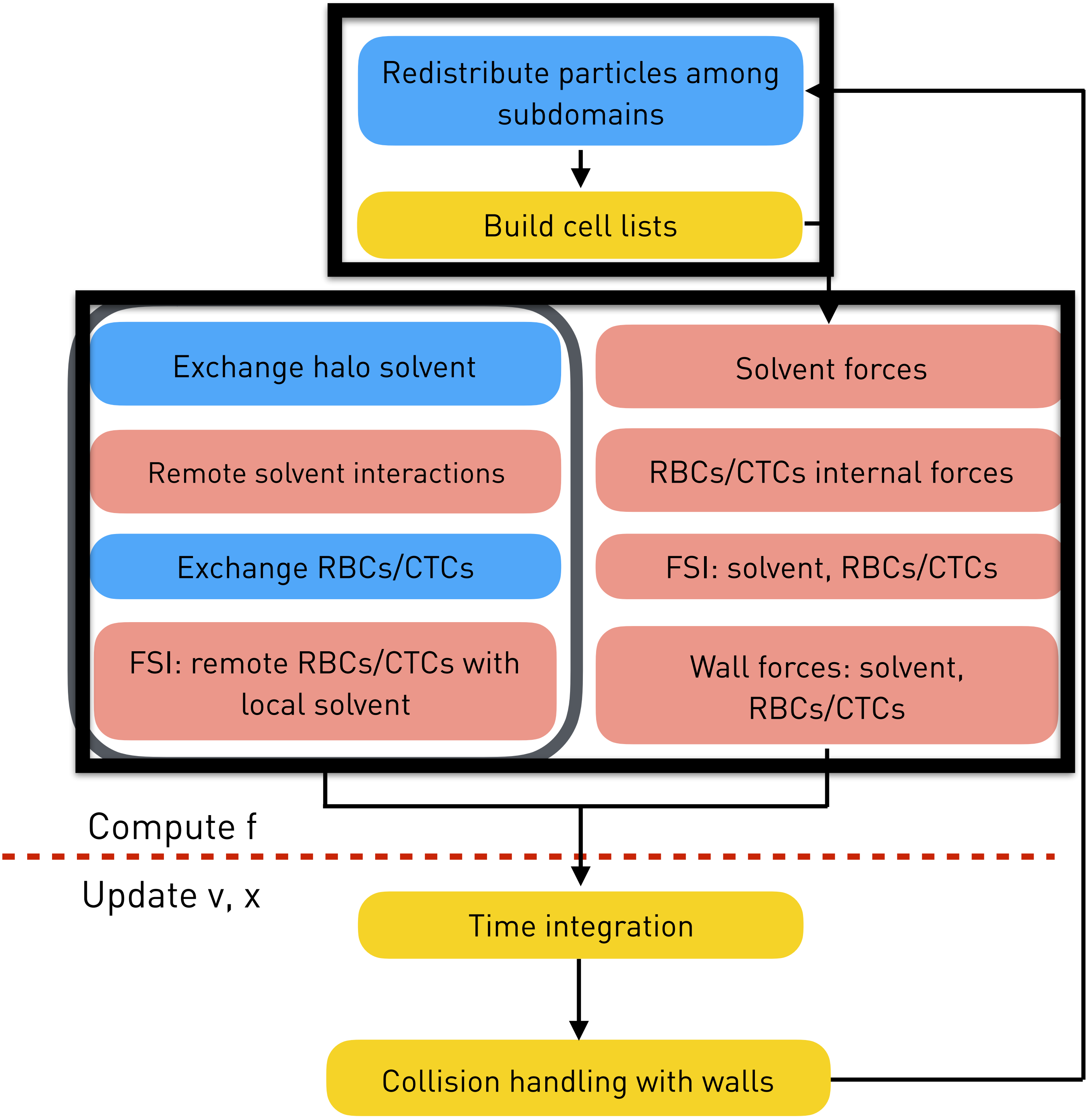
HOW TO MAXIMISE IPC

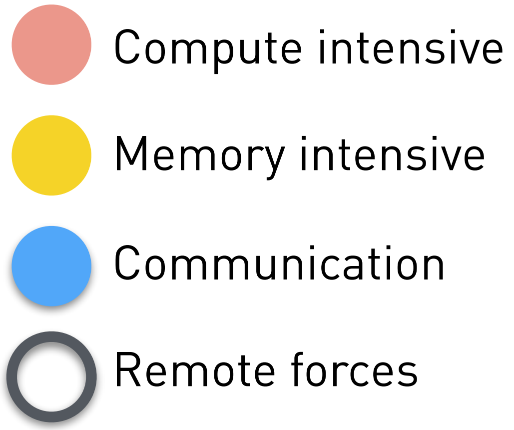


- ➡ Relax irregularities
- ➡ Balance the workload
- ➡ Prefer high-throughput instructions
- ➡ Exploit the parallelism in the system



NETWORK LEVEL





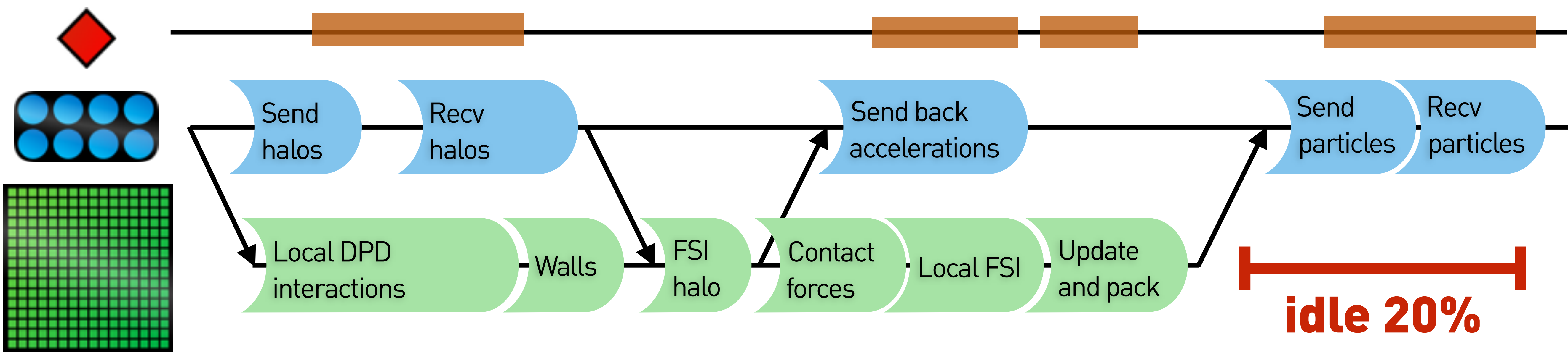
- Message sizes guessed a-priori
- Exceptions: secondary messages
- Adapted over recent history

➡ Non-blocking MPI calls

➡ C/T Overlap

NODE LEVEL

- Fully asynchronous CPU-GPU workflow
- Non-trivial dependency graph with 3 CUDA streams



Solution:

- Node oversubscription with multiple MPI tasks

➡ **GPU utilisation goes to over 95%**

➡ **Opportunities for load-balancing**

K20X FOR DPD SIMULATIONS

Throughput of Native Arithmetic Instructions.
(Number of Operations per Clock Cycle per Multiprocessor)

	Compute Capability					
	2.0	2.1	3.0, 3.2	3.5, 3.7	5.0, 5.2	5.3
16-bit floating-point add, multiply, multiply-add	N/A	N/A	N/A	N/A	N/A	256
32-bit floating-point add, multiply, multiply-add	32	48	192	192	128	128
64-bit floating-point add, multiply, multiply-add	16[1]	4	8	64[2]	4	4
32-bit floating-point reciprocal, reciprocal square root, base-2 logarithm (`__log2f`), base 2 exponential (`exp2f`), sine (`__sinf`), cosine (`__cosf`)	4	8	32	32	32	32
32-bit integer add, extended-precision add, subtract, extended-precision subtract	32	48	160	160	128	128
32-bit integer multiply, multiply-add, extended-precision multiply-add	16	16	32	32	Multiple instructions	Multiple instructions
24-bit integer multiply (`__[u]mul24`)	Multiple instructions	Multiple instructions	Multiple instructions	Multiple instructions	Multiple instructions	Multiple instructions
32-bit integer shift	16	16	32	64[3]	64	64
compare, minimum, maximum	32	48	160	160	64	64
32-bit integer bit reverse, bit field extract/insert	16	16	32	32	64	64
32-bit bitwise AND, OR, XOR	32	48	160	160	128	128
count of leading zeros, most significant non-sign bit	16	16	32	32	Multiple instructions	Multiple instructions

SP: best operation

DP: 3x penalty

OK

Integer multiplication: 6x penalty

OK

OK

➡ Maximize IPC

➡ **Maximize GPU throughput**

RANDOM NUMBER GENERATOR

- FMA-based RNG
- Passes BigCrush TestU01
- At least 18 rounds

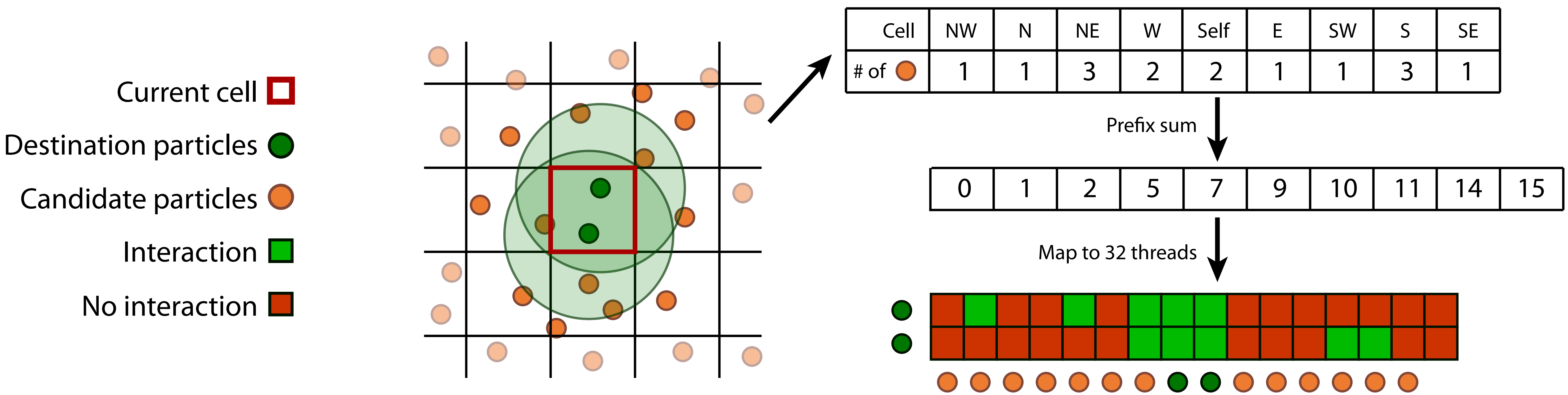
	SARU	OURS
FP32	30	**64**
non-FP32	81	16
TOT	111	**80**

```
function MEANOVAR1(i,j,k)
    // Low-discrepancy number
    u ← MIN(i,j)
    v ← MAX(i,j)
    p_ij ← MOD(u × G + v × S,1)
    y ← k − p_ij

    // Pass through Logistic map
    for N rounds do
        y ← 4y(1 − y)
    end for

    // Normalize
    z ← NORMALIZE(y)
    return z
end function
```

GPU LEVEL

Verlet lists:

- **Produced and consumed on-chip**
- Front-end: Verlet lists production
- Backend: DPD Interactions



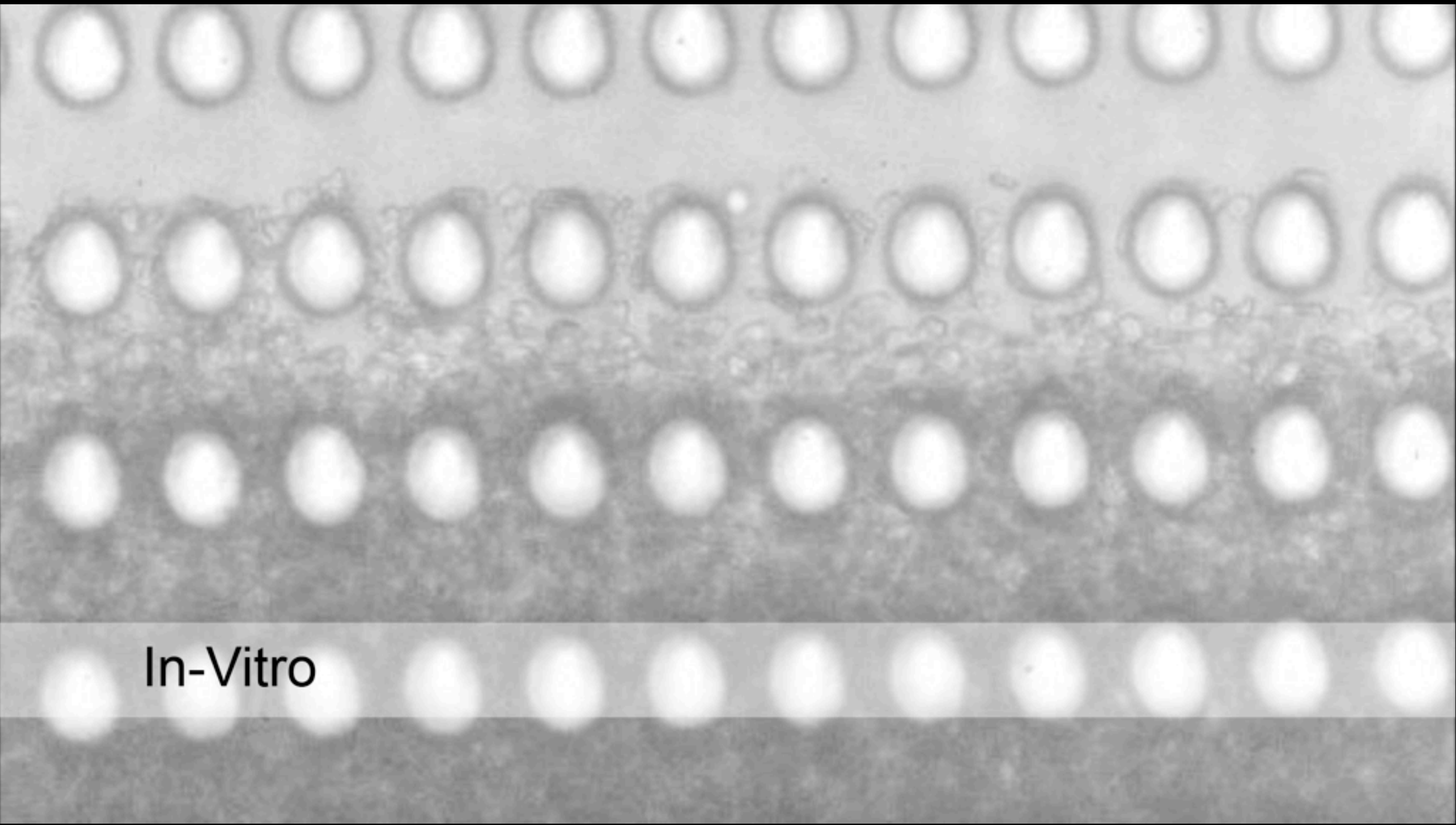
Workload of a cell-list dynamically mapped to a warp

- Latencies of the front-end hidden by the backend
- Warp dynamically configured (1x32, 2x16, 4x8)
- **SIMT far more flexible than SIMD here!**

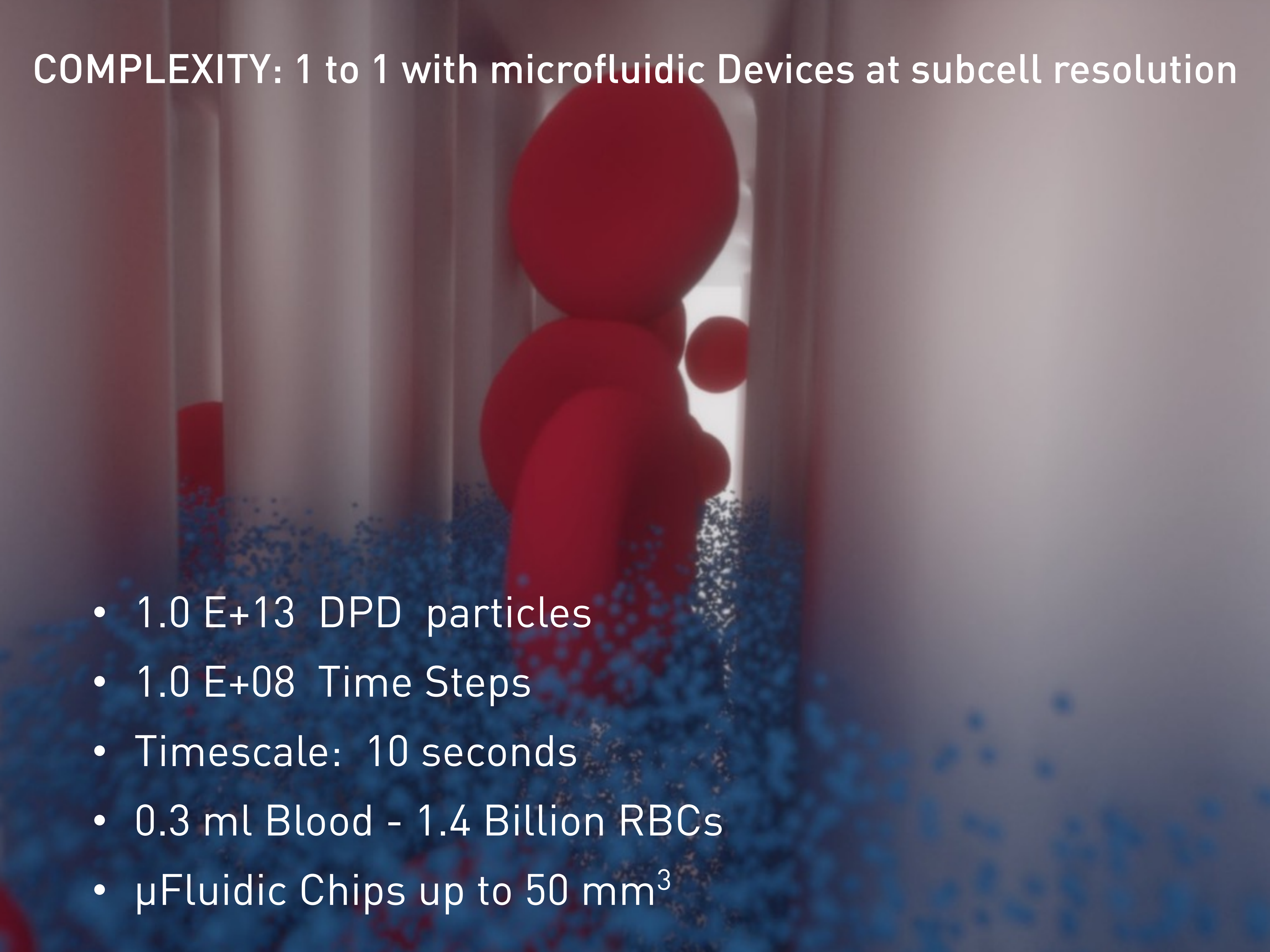
uDeviceX

http://udevicex.github.io/uDeviceX/

THE IN SILICO CTC-ICHIP



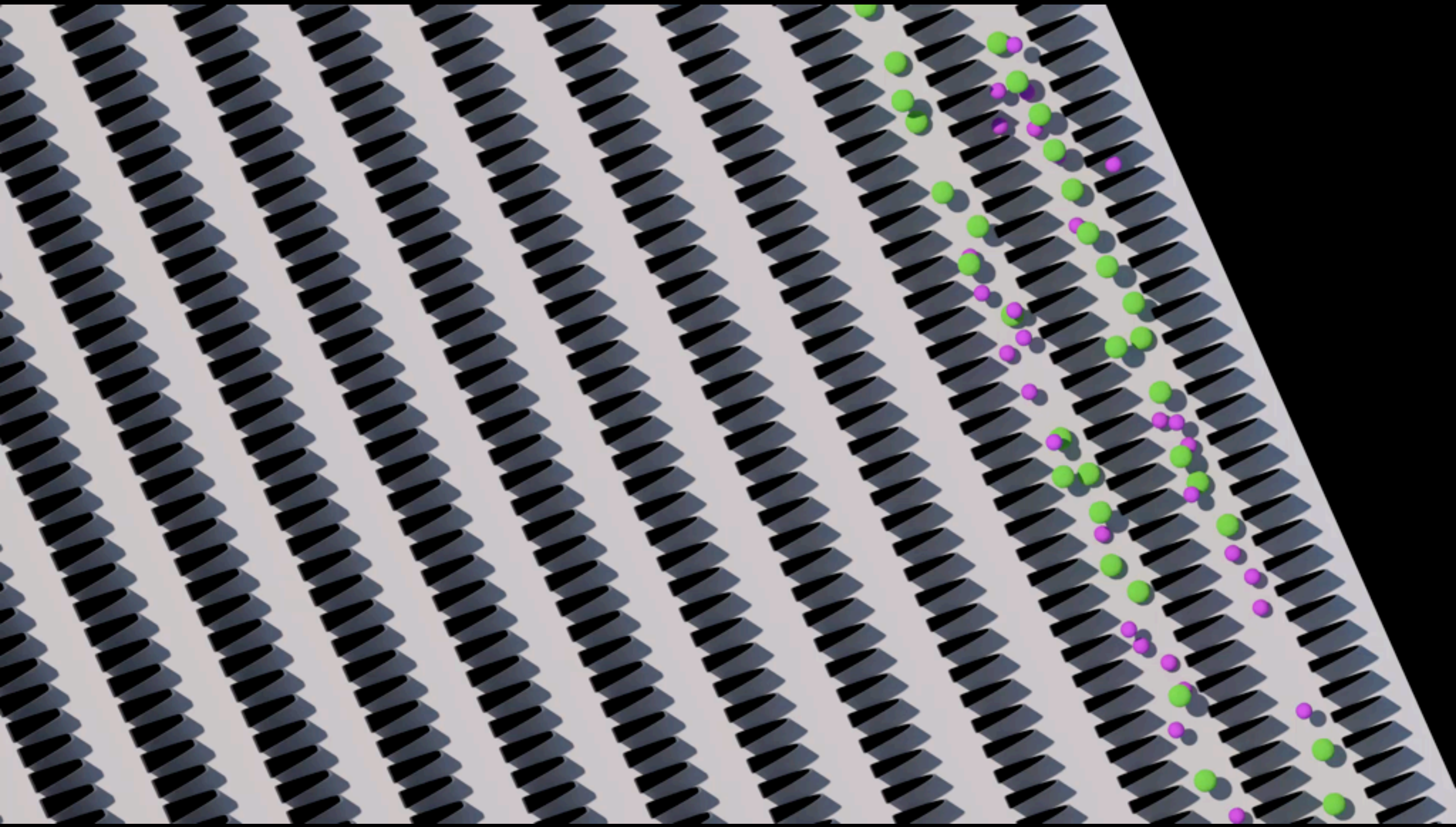
- TIME TO SOLUTION: >45X over State of the Art
- SCALING: >98% Weak and >87% Strong
- PERFORMANCE: max: 66% of nominal peak (avg. 34%)



COMPLEXITY: 1 to 1 with microfluidic Devices at subcell resolution

- 1.0 E+13 DPD particles
- 1.0 E+08 Time Steps
- Timescale: 10 seconds
- 0.3 ml Blood - 1.4 Billion RBCs
- μFluidic Chips up to 50 mm^3

FUNNEL RATCHETS DEVICE



ACKNOWLEDGMENTS

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BROWN

Università della Svizzera italiana

Consiglio Nazionale delle Ricerche

OAK RIDGE
National Laboratory

U.S. DEPARTMENT OF ENERGY
INCITE
LEADERSHIP COMPUTING

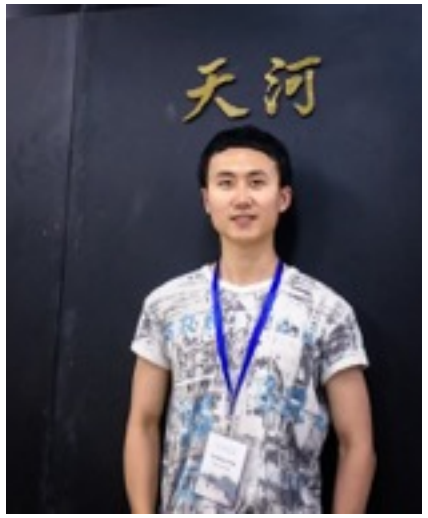
NVIDIA

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

NIH
National Institutes of Health

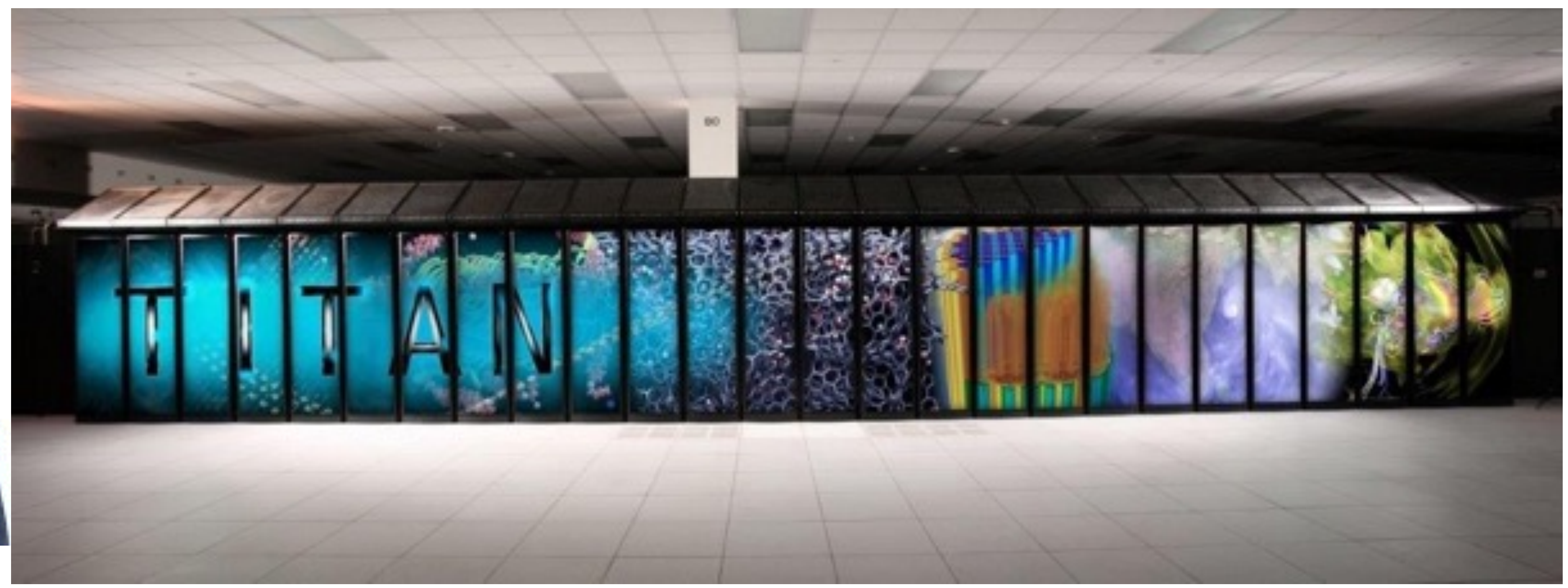
FNSNF
Fonds national suisse
Schweizerischer Nationalfonds
Fondo nazionale svizzero
Swiss National Science Foundation

erc
European Research Council
Established by the European Commission


天河






TITAN



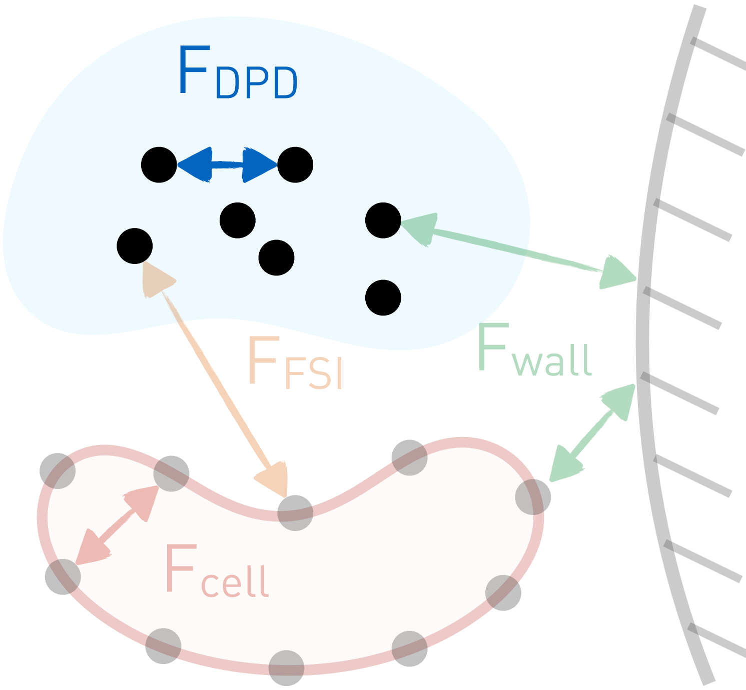

ELEMENT

THE END

Thank you!

DISSIPATIVE PARTICLE DYNAMICS

- Models the solvent
- Short-range interactions
- Fluctuation-dissipation theorem
- Correct hydrodynamics for Re < 10



$$\mathbf{F}_i = \sum_{j \neq i}^{N} (\mathbf{F}_{ij}^C + \mathbf{F}_{ij}^D + \mathbf{F}_{ij}^R), \qquad i = 1, \ldots, N$$

Conservative:

$$\mathbf{F}_{ij}^C = a w(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}}$$

Random:

$$\mathbf{F}_{ij}^R = \sigma w(r_{ij}) \theta \frac{\mathbf{r}_{ij}}{r_{ij}}$$

Kernel

Dissipative:

$$\mathbf{F}_{ij}^D = -\gamma (w(r_{ij}))^2 (\mathbf{r}_{ij}, \mathbf{u}_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}^2}$$

[Hoogerbrugge and Koelman, 1992, EPL] [Groot and Warren, 1997, JCP] [Español, 1995, PRE]

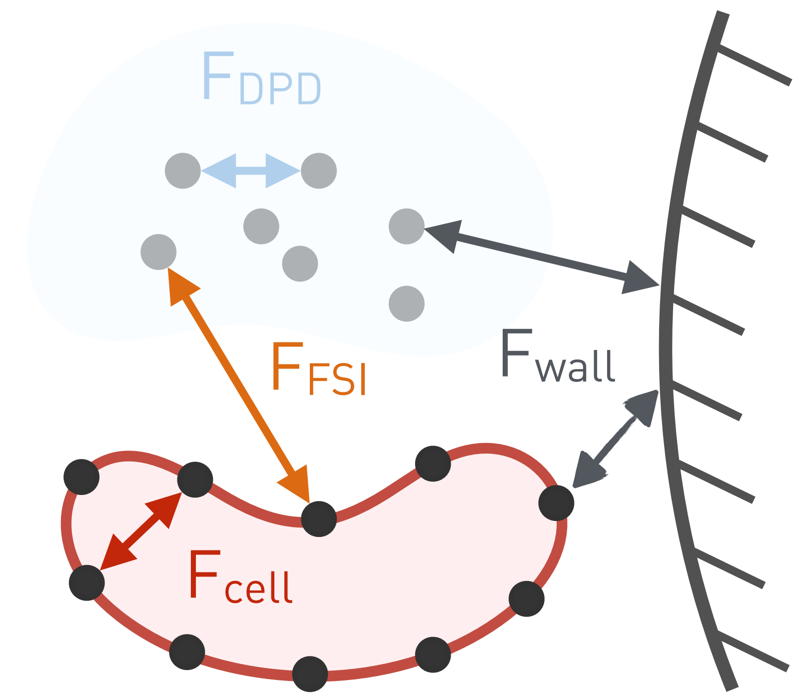
CELLS AND FLUID-STRUCTURE INTERACTIONS

Cells:

- Deformable membrane model
- Discretised as triangle mesh
- Membrane forces are stiff

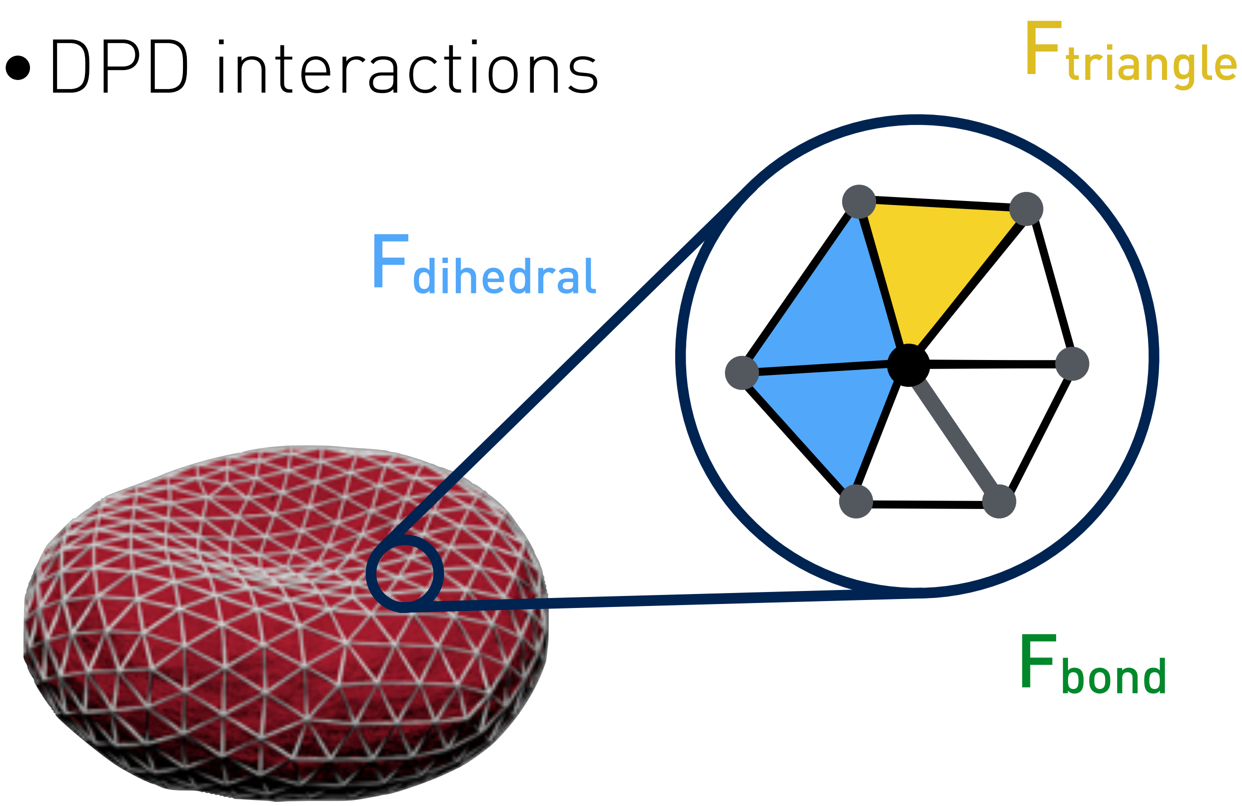
Walls:

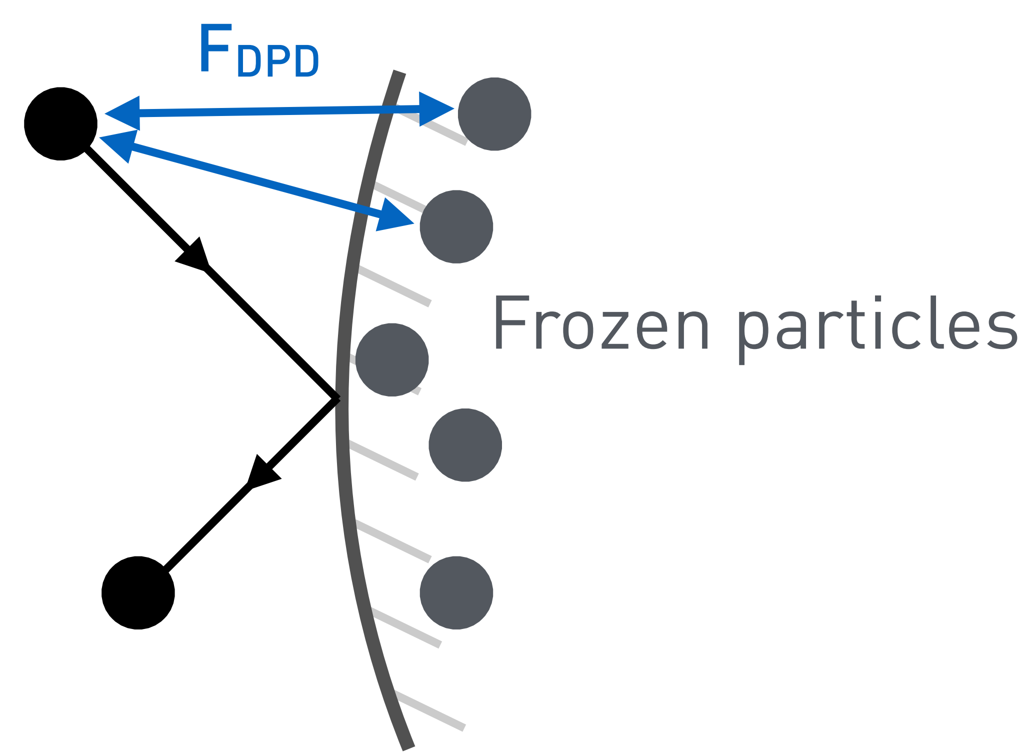
- Implicit description
- Frozen particles
- Bounce-back



FSI forces:

- DPD interactions

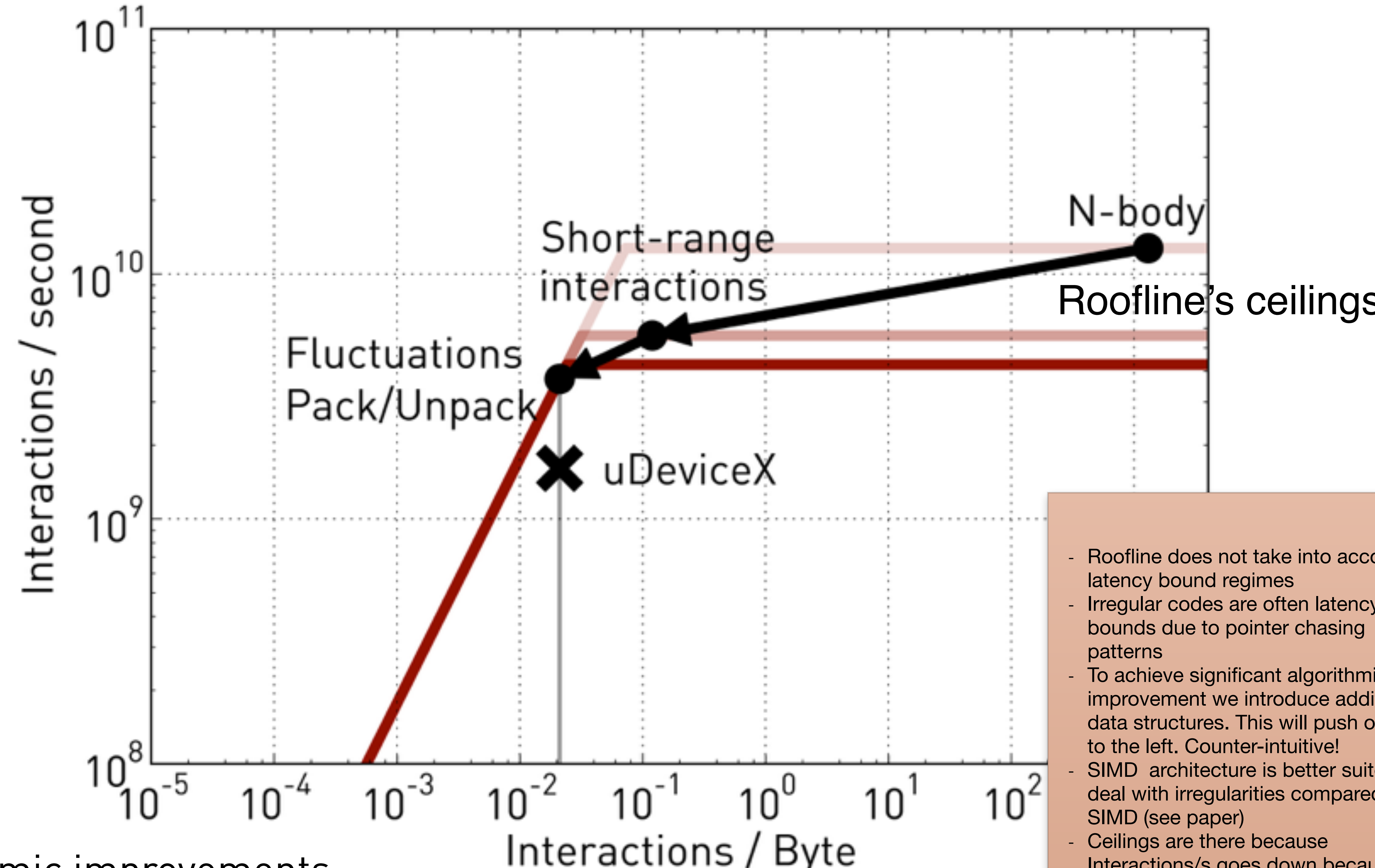




[Li et al., 2005, BPJ] [Pan, Pivkin and Karniadakis, 2008, EPL]

[Visser, Hoefsloot and Iedema, 2005, *JCP*]

THE DOWNSIDE OF ALGORITHMIC IMPROVEMENTS

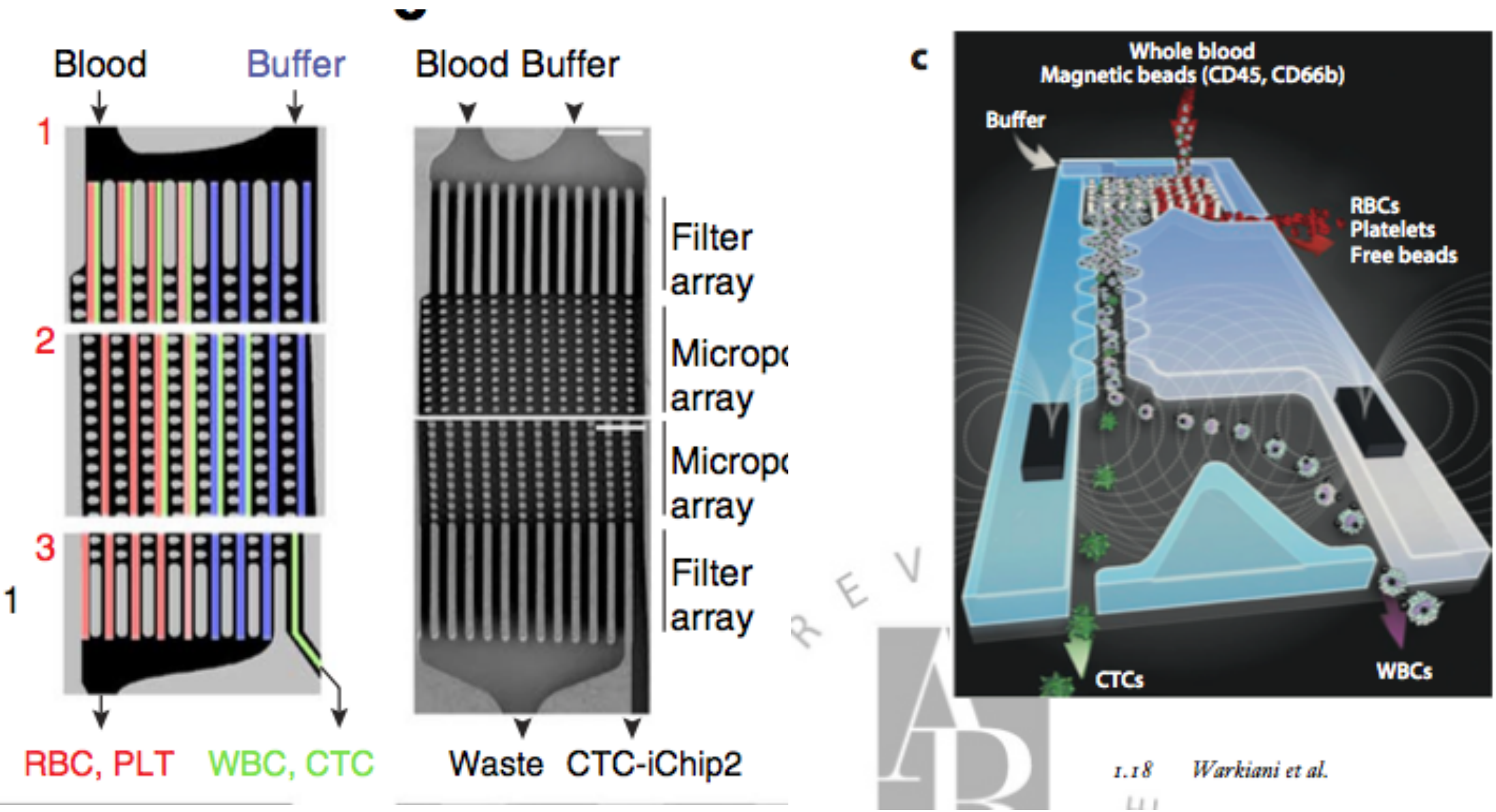


Algorithmic improvements

- Lead to irregular codes
- Often enters in a latency-bound regime
- OIs shift to the left

- Roofline does not take into account latency bound regimes
- Irregular codes are often latency bounds due to pointer chasing patterns
- To achieve significant algorithmic improvement we introduce additional data structures. This will push our OI to the left. Counter-intuitive!
- SIMD architecture is better suited to deal with irregularities compared to SIMD (see paper)
- Ceilings are there because Interactions/s goes down because we need to spend instructions and time somewhere else (cell lists, packing unpacking etc)

CTC-ICHIP

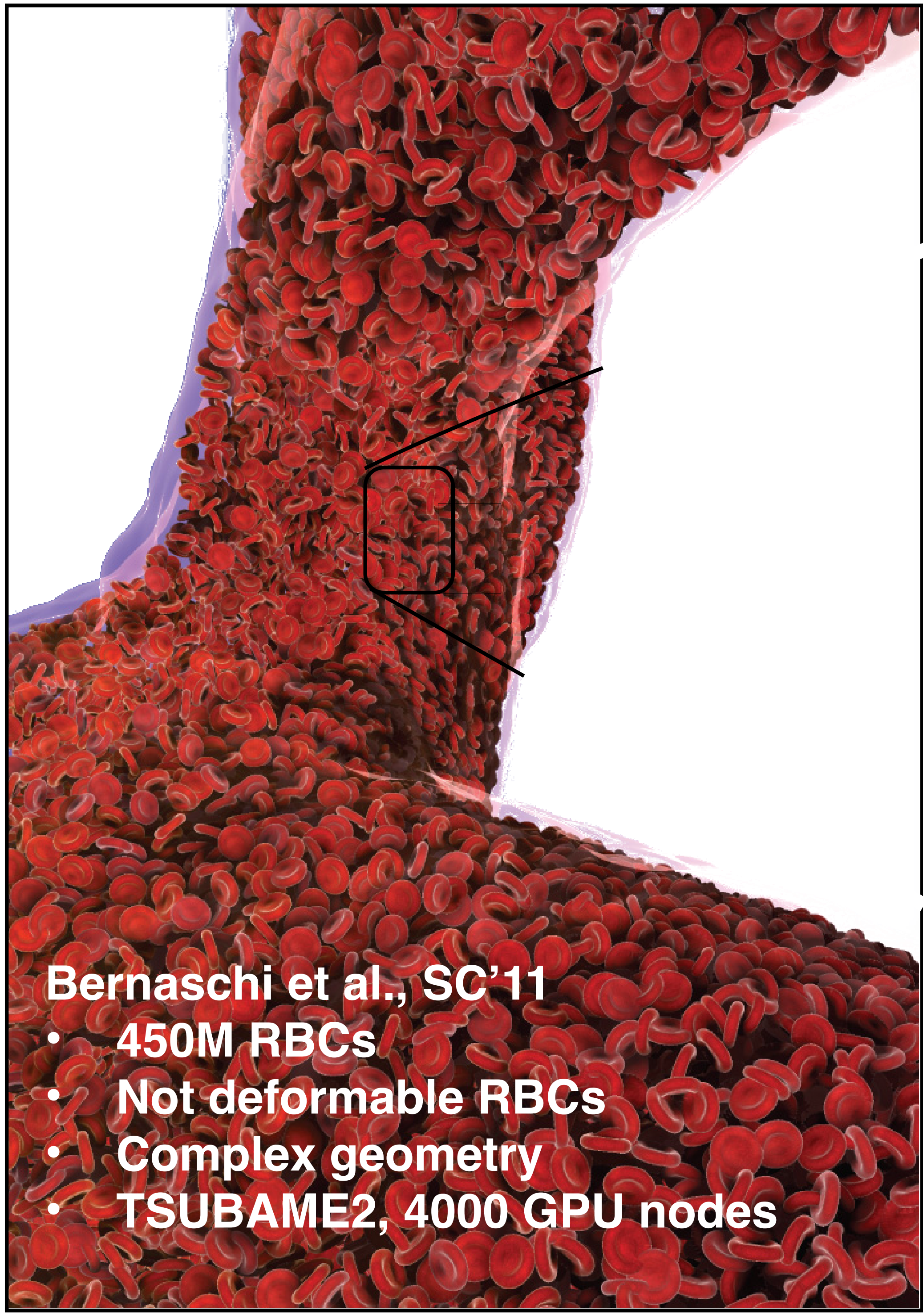

Blood
Buffer
1
2
3
RBC, PLT
WBC, CTC
Blood Buffer
Filter array
array
array
Filter array
Waste
CTC-iChip2
c
Whole blood
Magnetic beads (CD45, CD66b)
Buffer
RBCs
Platelets
Free beads
CTCs
WBCs
1.18 Warkiani et al.

COMPARISON WITH STATE-OF-THE-ART

Reference work:

- $7.3 * 10^{10}$ unknown/s on TSUBAME 2
- $1.0 * 10^{12}$ unknown/s on TITAN assuming perfect scaling

➡ **uDeviceX: 7.3X** higher throughput



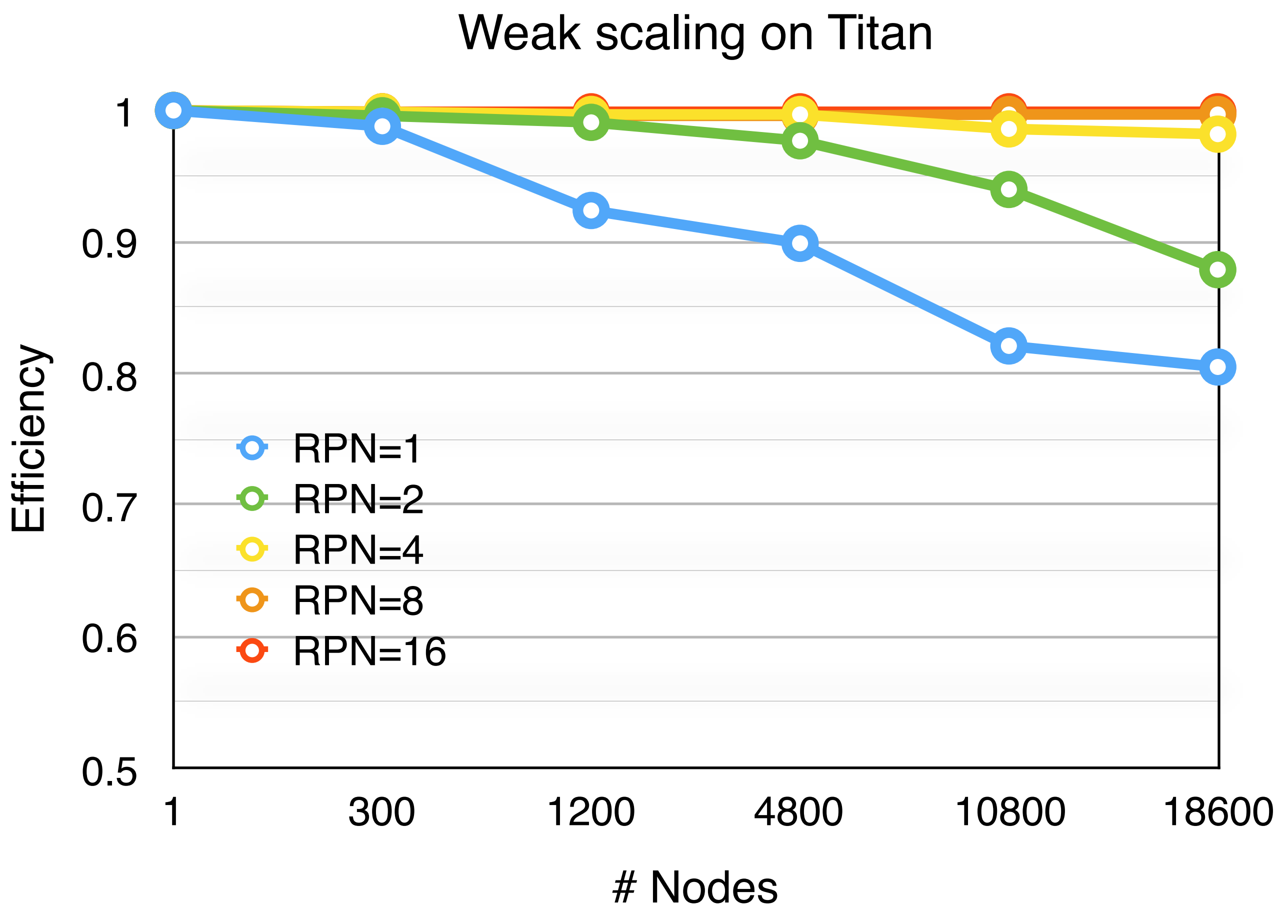
SCALABILITY

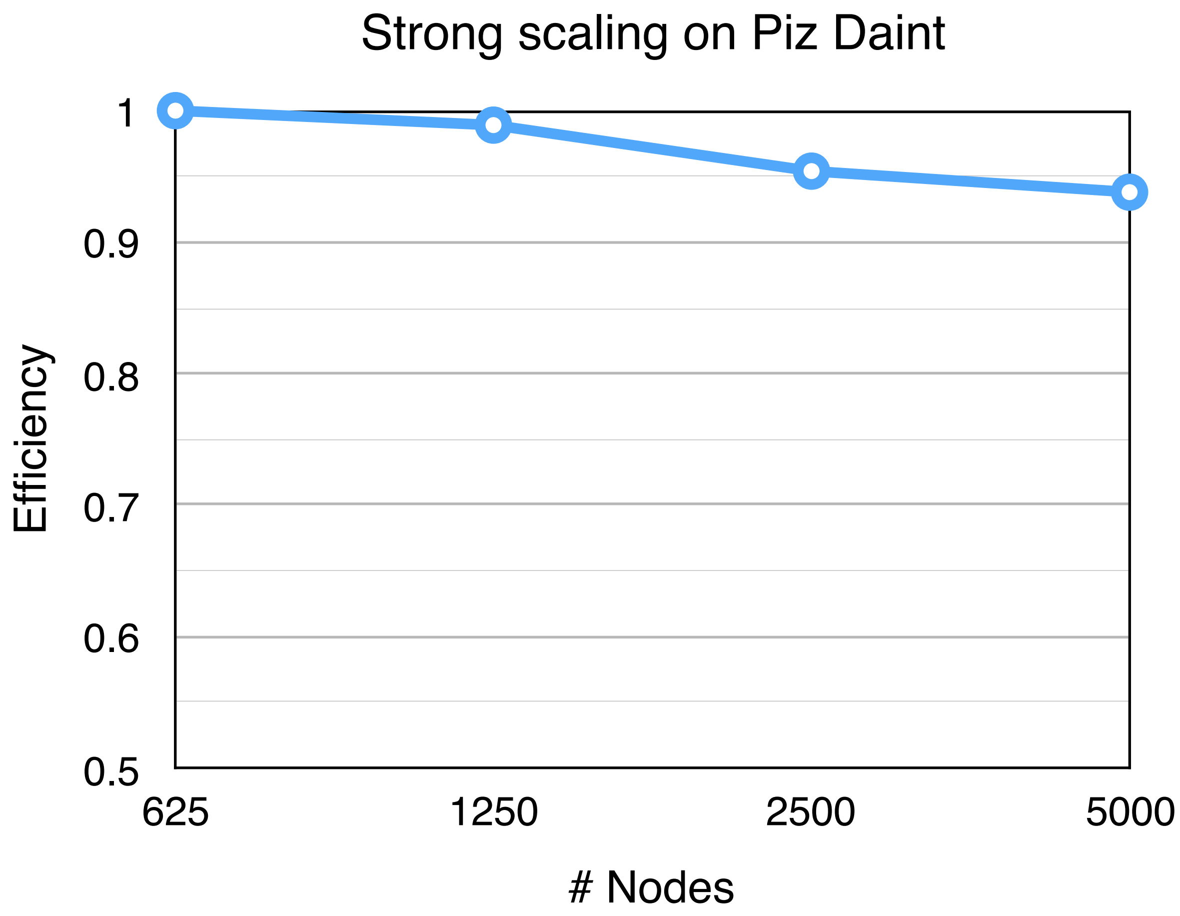
Do we scale? YES!

Some teams compete for SCALABILITY.

> **98% weak scaling** on the 18k nodes of Titan

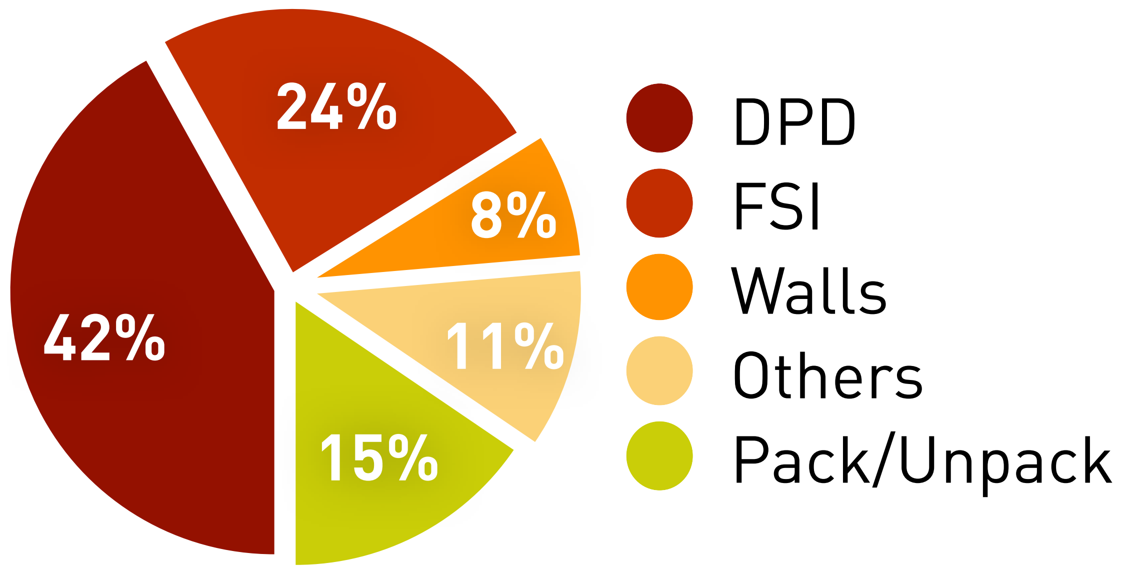
94% strong scaling from 625 to 5000 nodes of Piz Daint





FURTHER ANALYSIS

- **CUDA kernels**: 85% of the GPU time

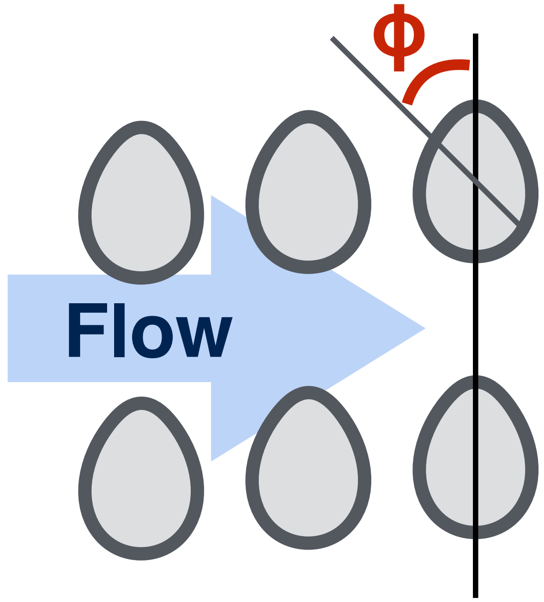


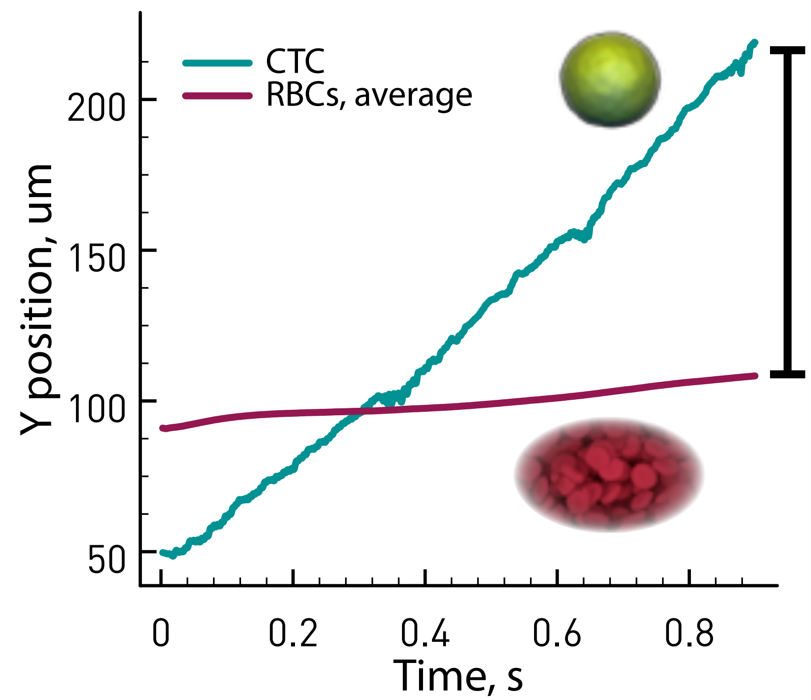
Kernel	IPC	% GPU peak
DPD	2.7	45%
FSI	2.5	43%
Walls	3.2	52%

- **Time scales separation**
  - 5.5x gain in time-to-solution
  - Increased GPU utilization: IPC = 40% of the nominal peak performance

OPTIMIZING CTC-ICHIP

Original CTC-iChip:

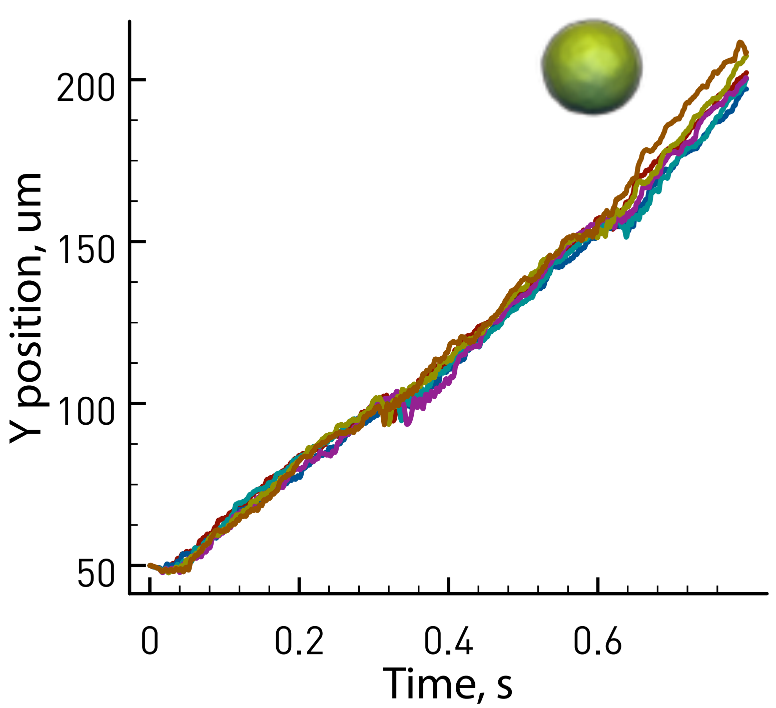




Optimizing for separation of CTC and RBCs

Optimizing **angle ϕ**

CTC trajectories are similar for different angles



RBCs behave very differently