OpenCL vs: Accelerated Finite-Difference Digital Synthesis

Harri Renney & Benedict R. Gaster

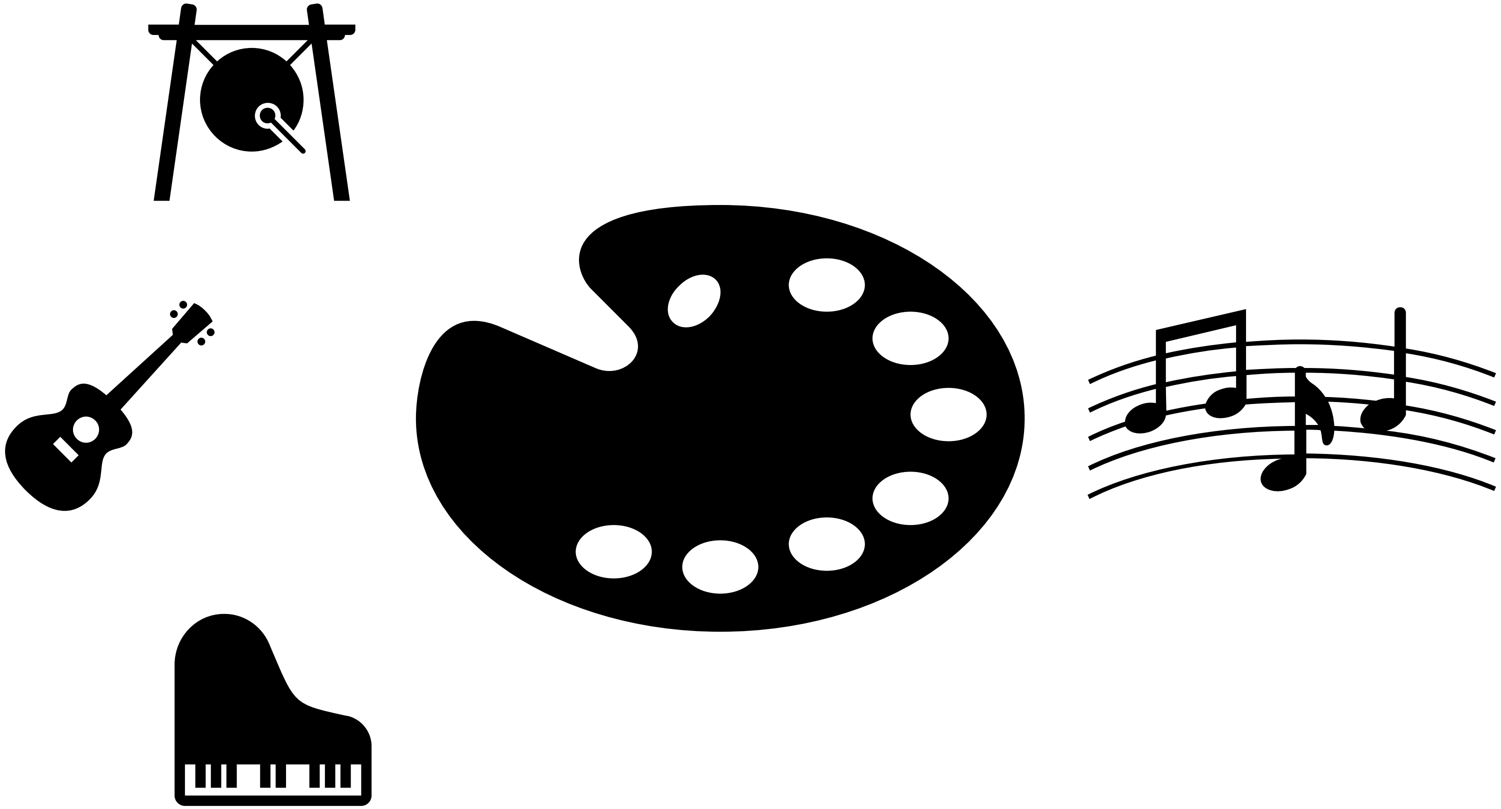
Computer Science Research Centre

Tom Mitchell

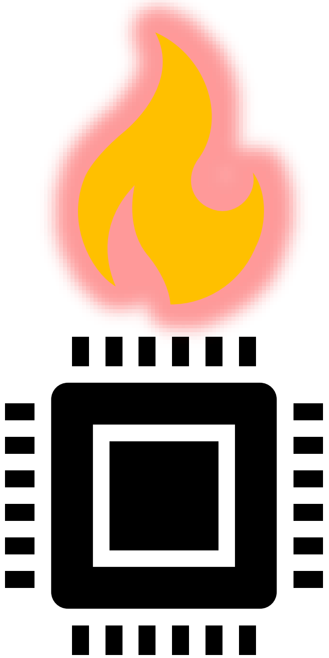
Computer Science and Creative Technologies

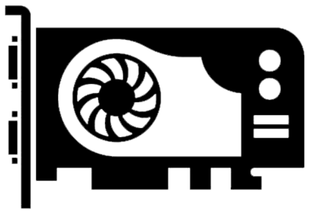
Motivation

Quality Sounds



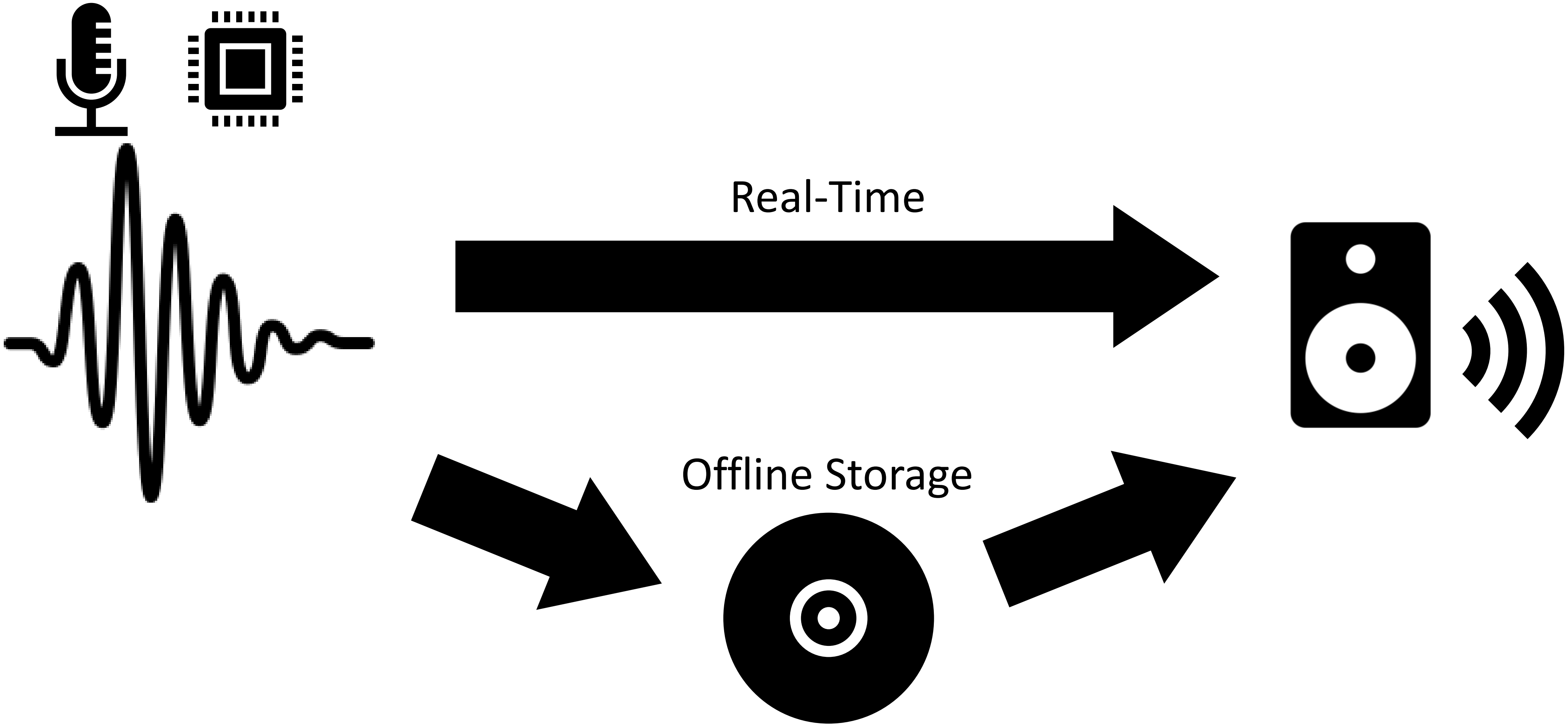
Intensive Music Production





Background

Digital Audio



Physical Modelling

Simple 1D String Physical Model: Karplus and Strong’s String Synthesis

2D Vocal Tract: Digital Waveguide Synthesis

Physical Modelling: Direct Numerical Simulation

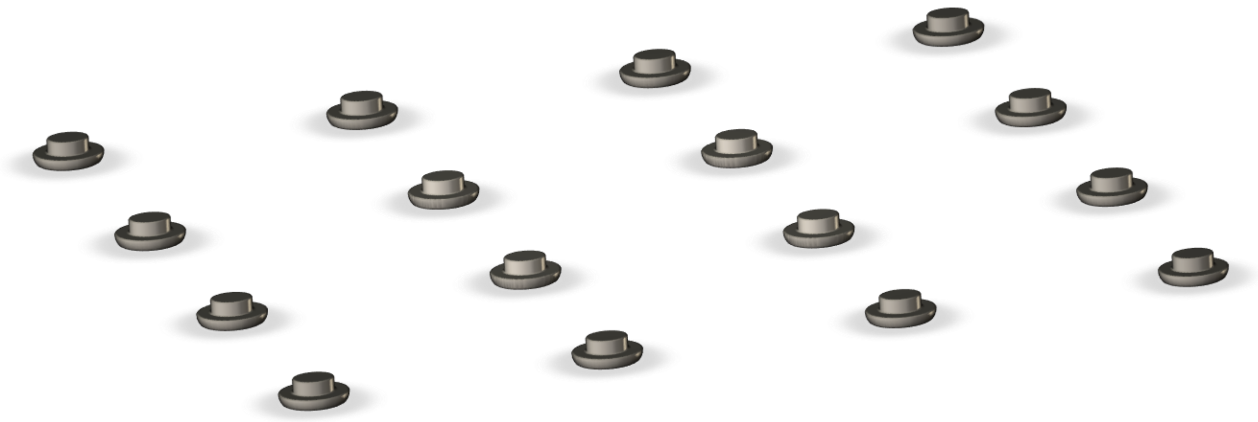
Applied Mathematics

Computer Program

Finite-Difference Time-Domain

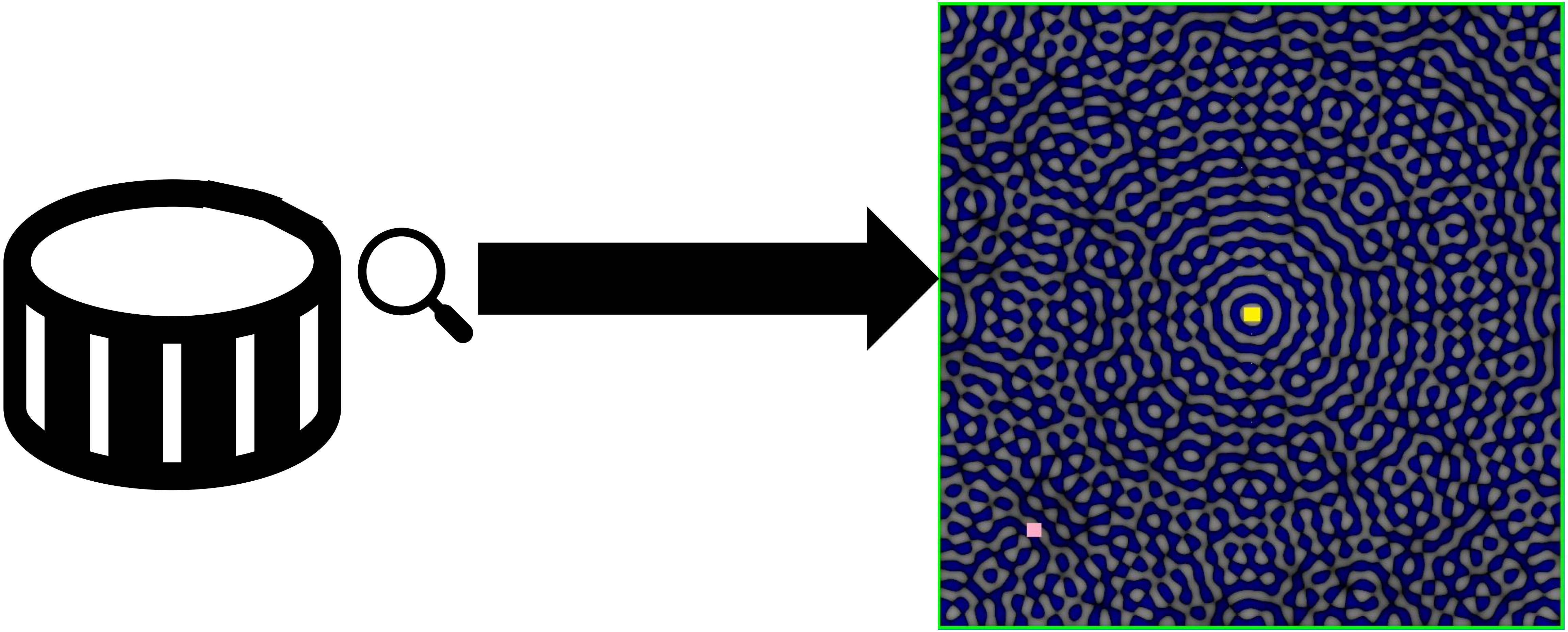
Finite difference methods represent differential equations.

Time-stepping creates simulation over time.



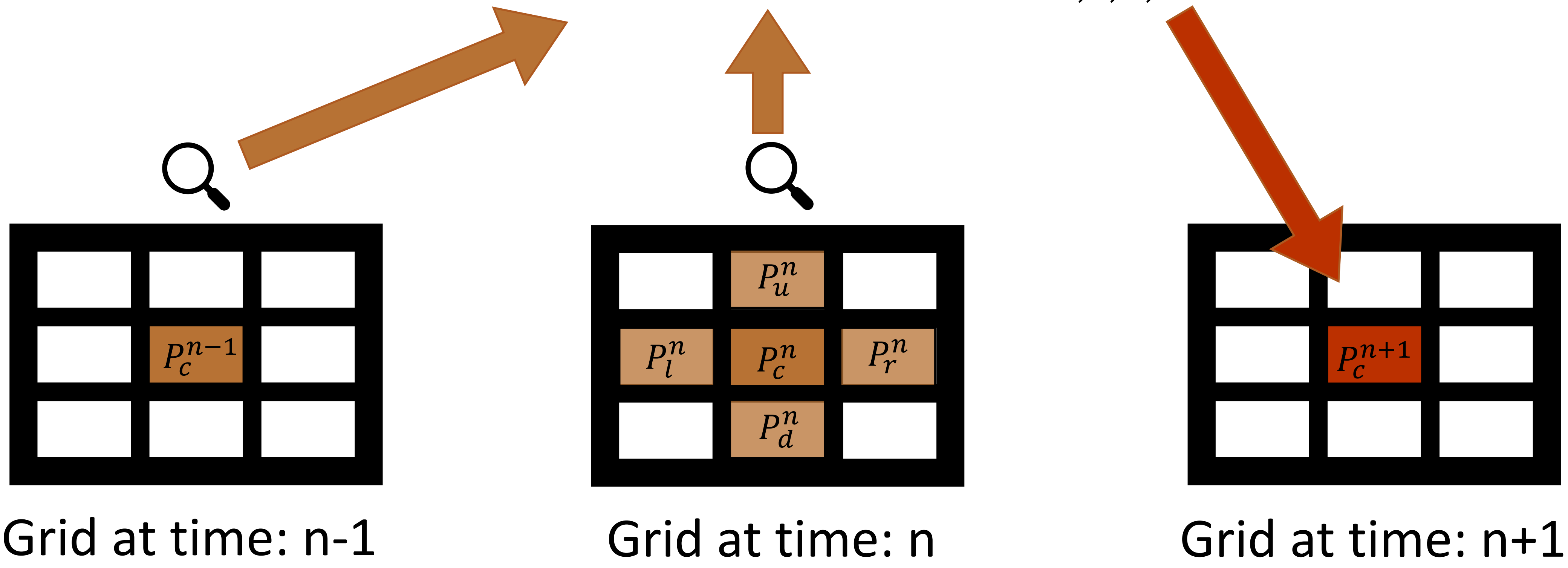
Design & Implementations

Physical Model: 2D Drumhead Membrane



General Architecture

$$P_c^{n+1} = calculate(P_c^{n-1}, P_c^n, P_{l,r,u,d}^n)$$



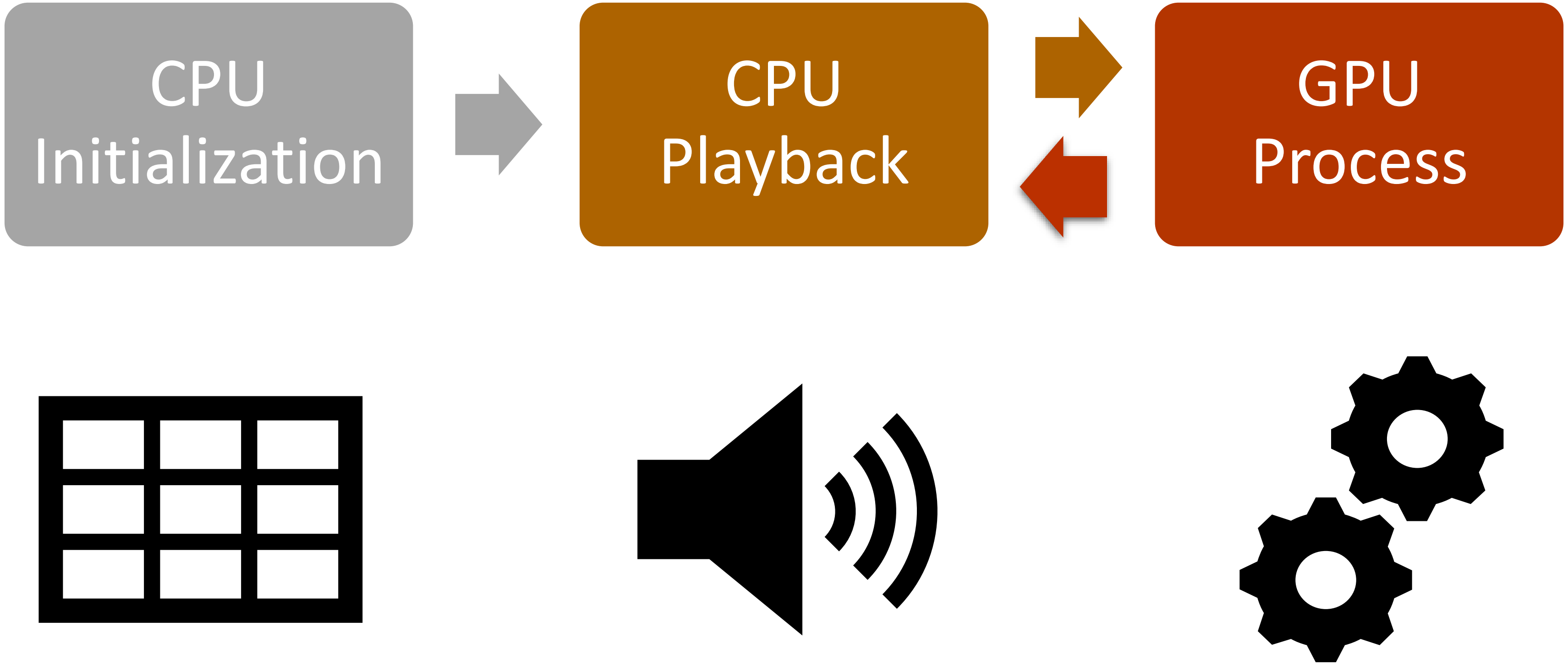
The Difference Equation

$$p^{n+1} = \frac{2p^n + (\mu - 1)p^{n-1} + \alpha(p_l + p_r + p_u + p_d - 4p^n)}{\mu + 1}$$ (Walstijn & Kowalczyk, 2008)

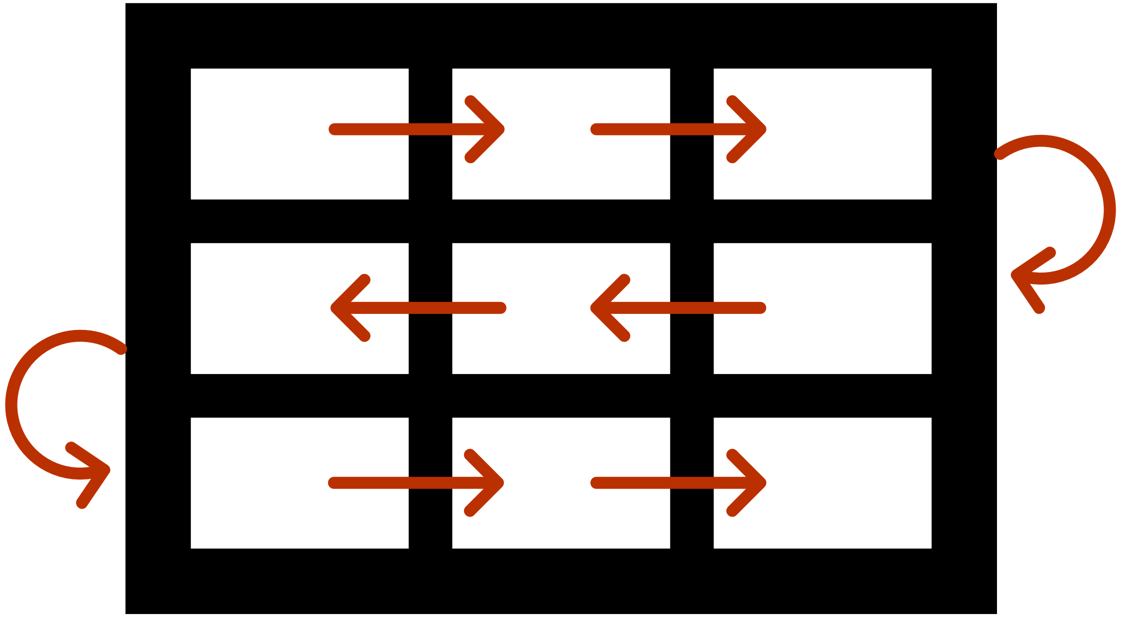
$$P_{L,R,U,D} = \begin{cases} p^n \gamma & \textit{if n boundary} \\ p^n_{l,r,u,d} & \textit{if n free} \end{cases}$$

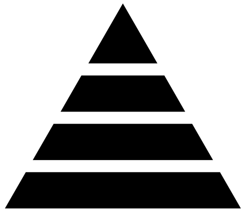
p^{n+1} = Pressure next time step
p^n = Pressure current time step
p^{n-1} = Pressure previous time step
μ = Damping coefficient $(0.0 < \mu < 1.0)$
α = Propagation coefficient $(\alpha \leq 0.5)$
$p_{l,r,u,d}$ = Pressure neighbours
γ = Boundary gain $(0.0 < \gamma < 1.0)$

GPU Acceleration

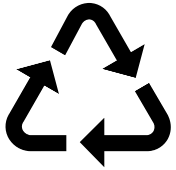


Version: CPU Serial

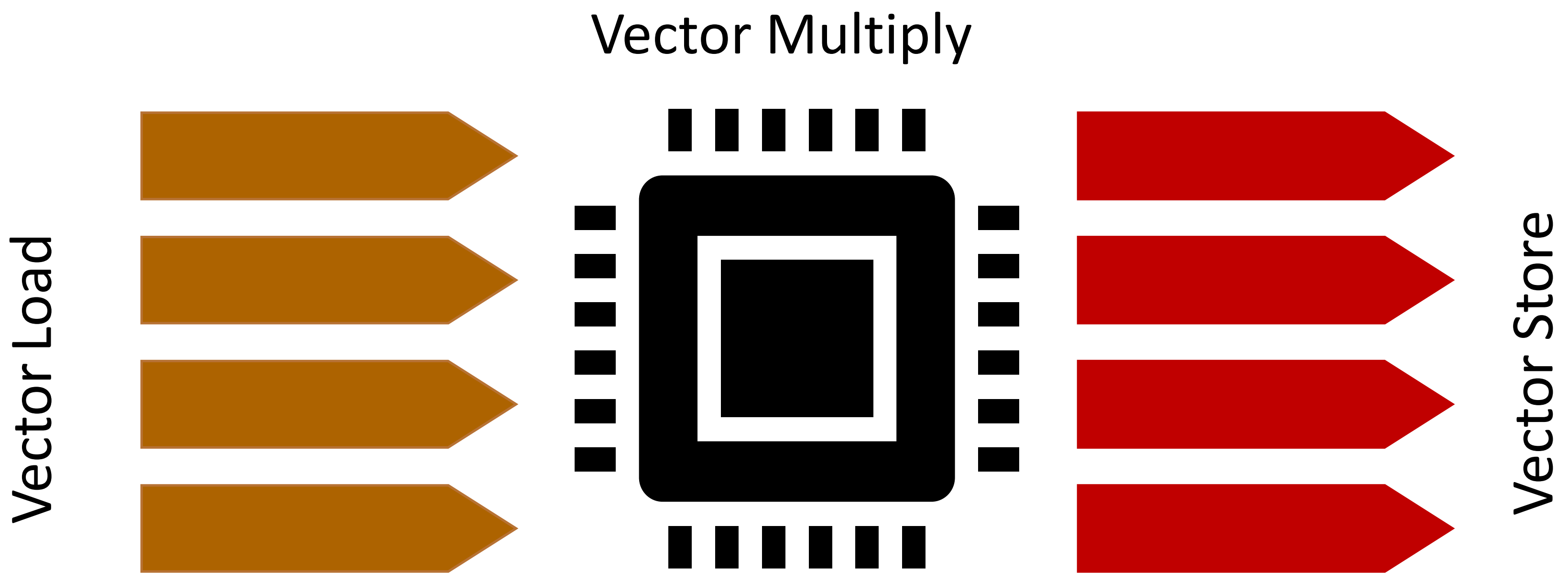


Cache Alignment: 

Avoiding Copies: 

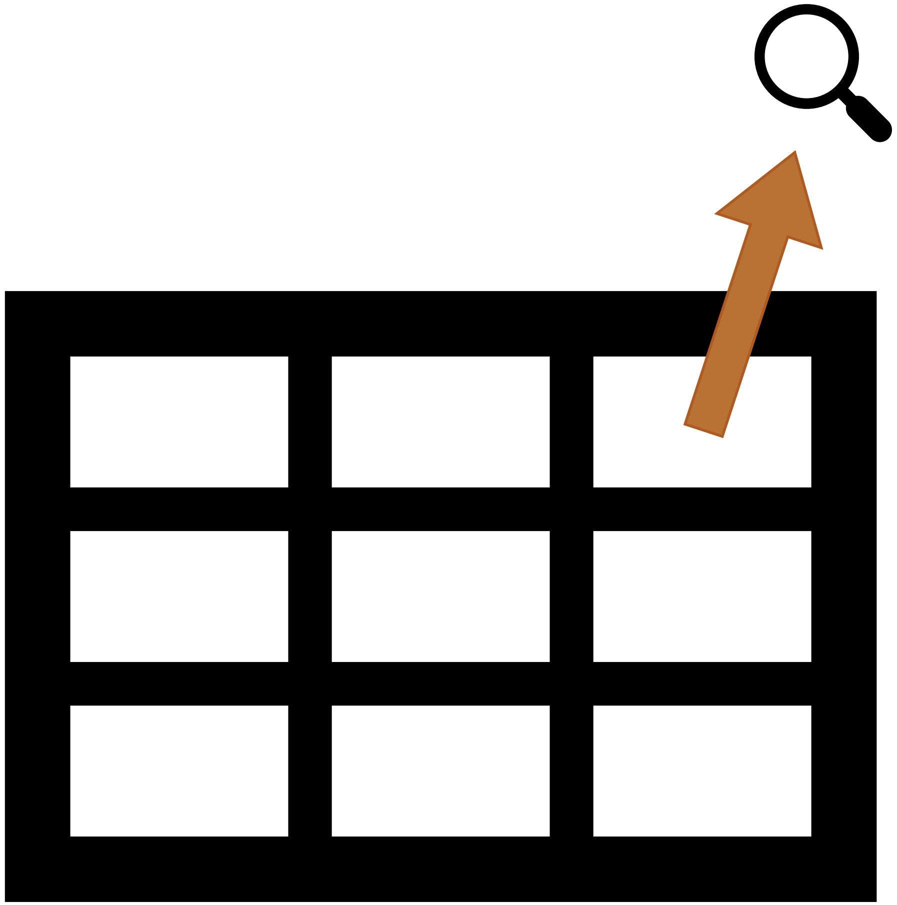
Redundant Calculations: 

Version: CPU AVX



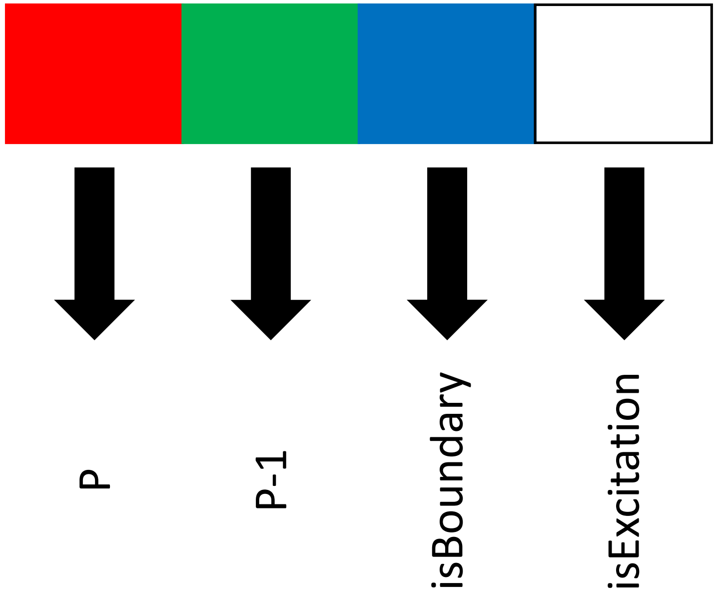
Version: GPU OpenGL

Design proposed by (Zappi et al, 2017)



Model stored as texture

Pixel: RGBA Channel



Version: GPU OpenCL

General Purpose GPU (GPGPU)
(Harris, 2012)

OpenCL

Global Workspace (Kernel)

Local Workspace (Workgroup)

Private Workspace (Work-item)

Version: OpenCL Global & Local

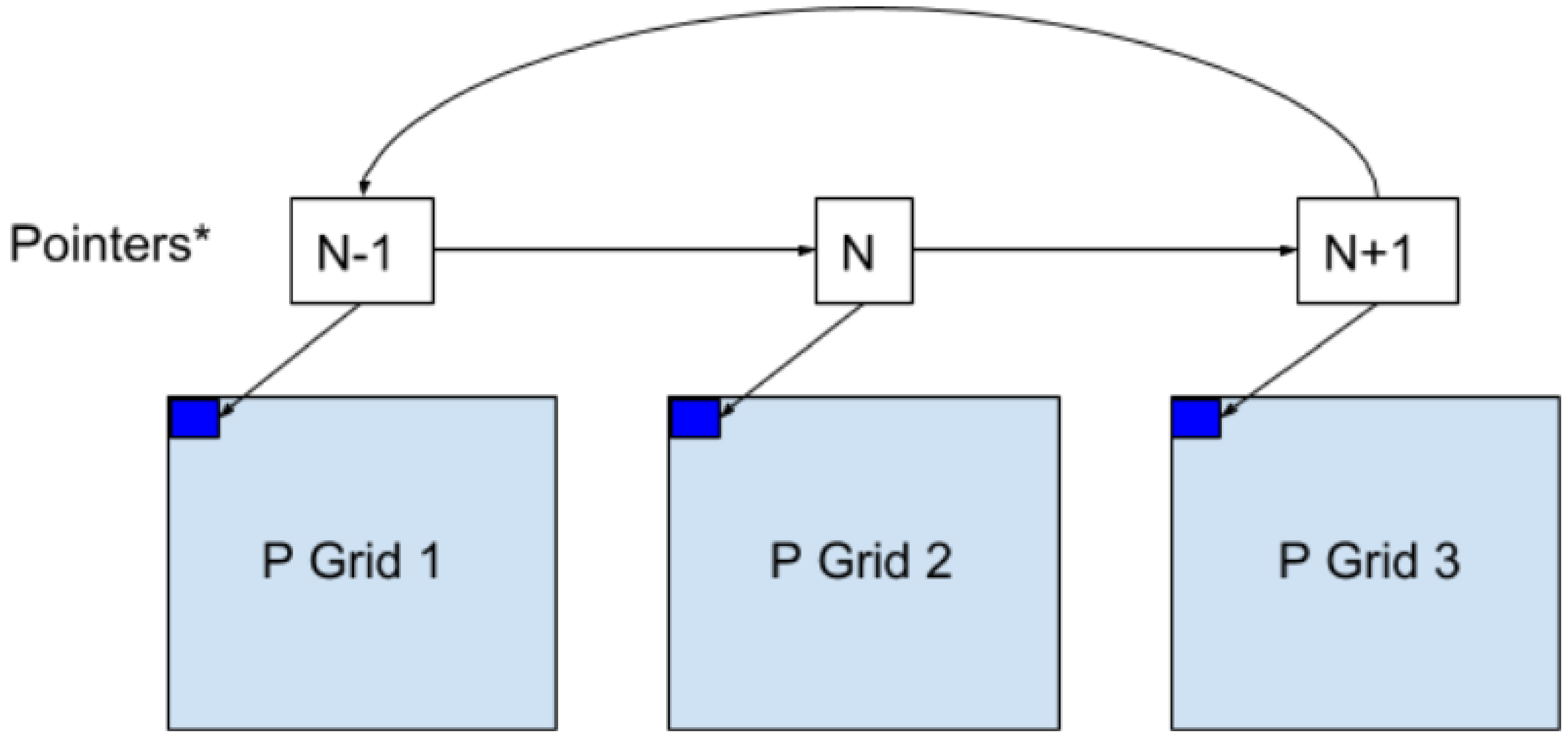
Purely global memory version

Local memory version

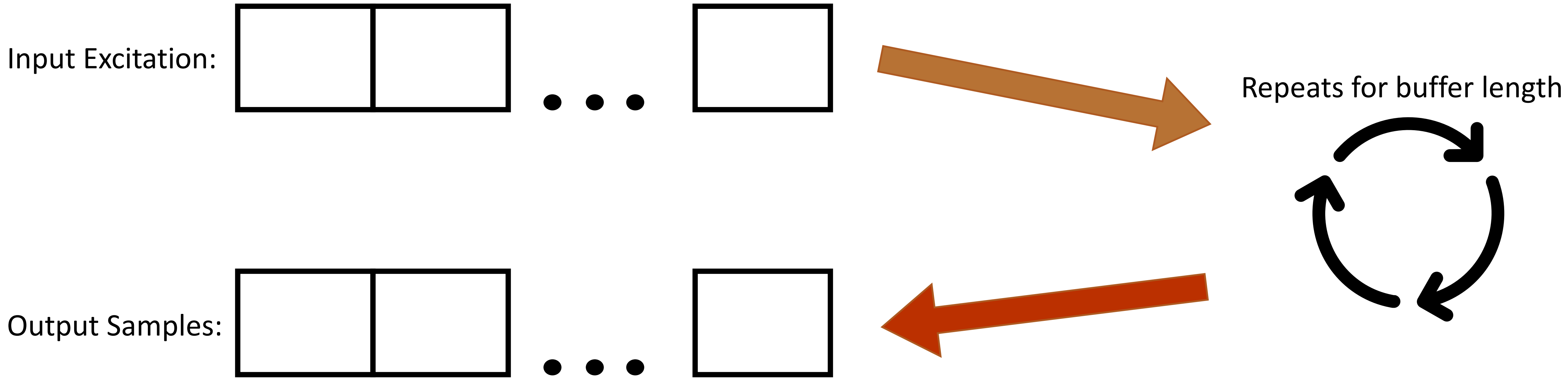
Version: Overview

FDTD Synthesis Implementations
CPU Serial
CPU AVX
GPU OpenGL
GPU OpenCL Global
GPU OpenCL Local

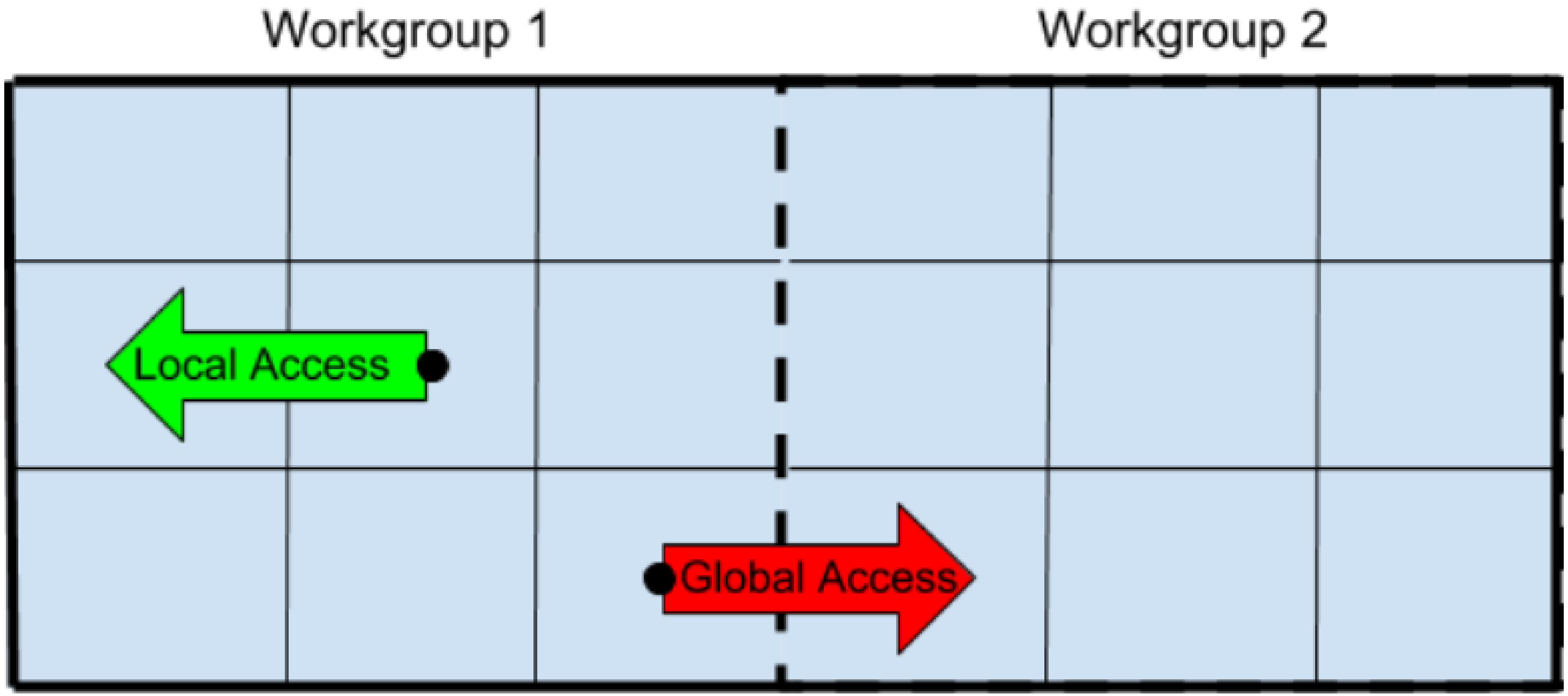
Common Mechanism: Rotation Index



Common Mechanism: Input/Output Buffer



OpenCL Workgroup Optimizations



Benchmarking

Control Parameters: Input/Output Buffer Size = 512

Independent Variables: Grid Dimensions, Workgroup Dimensions

Dependant Variables: Computation Time

System Specifications	
CPU	Intel Core i7-8550U 4 Cores 1.99GHZ
GPU	AMD Radeon 530 with 2GB GDDR5
CPU RAM	8GB 2400MHz DDR4

Results: Workgroup Dimensions

Compute time up to wavefront size

	OpenCL Global	OpenCL Local
Workgroup Size	time (ms)	time (ms)
4x4	136.072666	149.841333
8x8	46.170966	49.19716
16x16	46.5292	45.363366

Time per 512 samples.

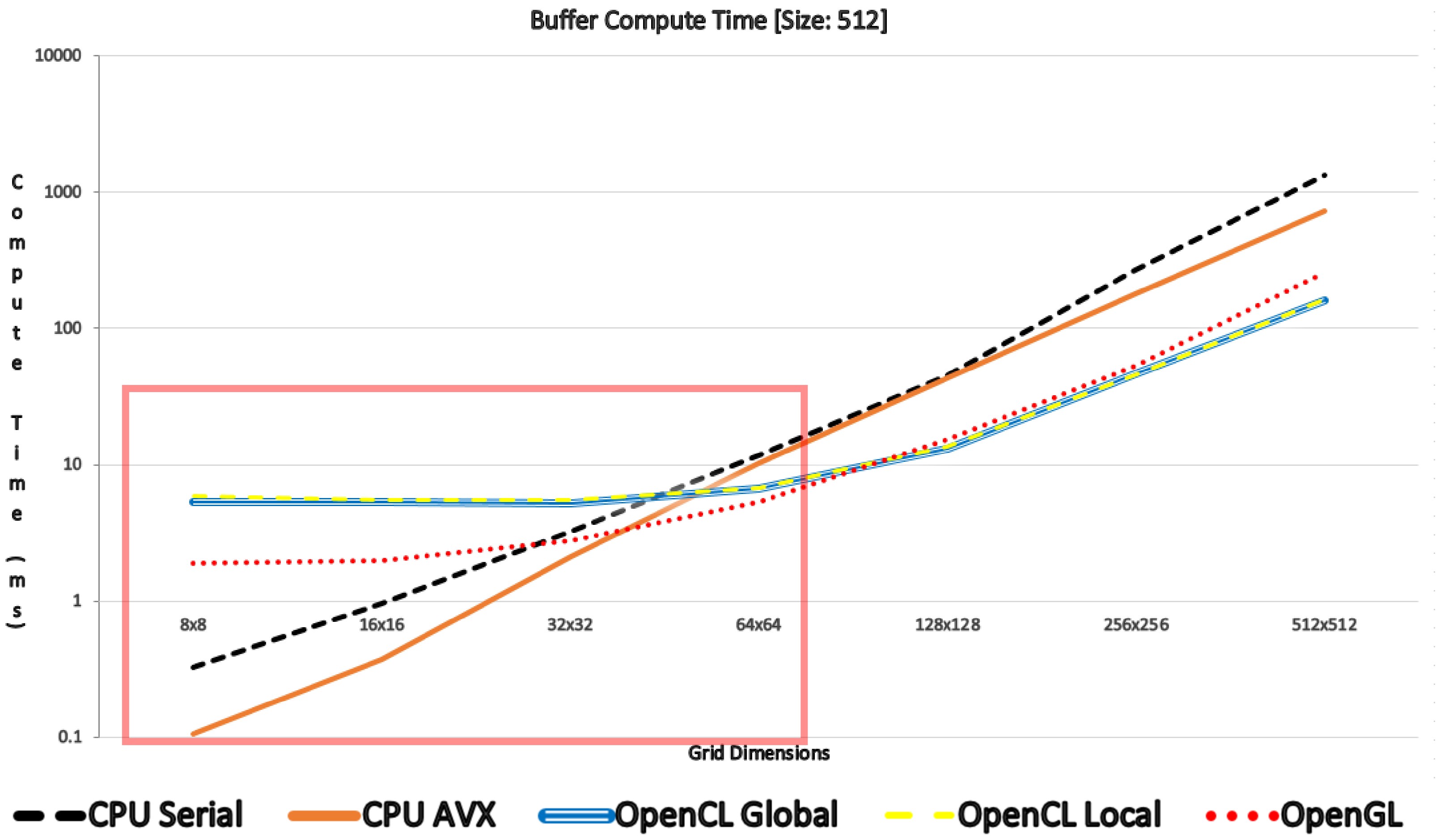
Compute time past wavefront size

	OpenCL Global	**OpenCL Local**
Workgroup Size	time (ms)	time (ms)
4x4	136.072666	149.841333
8x8	46.170966	49.19716
16x16	46.5292	45.363366

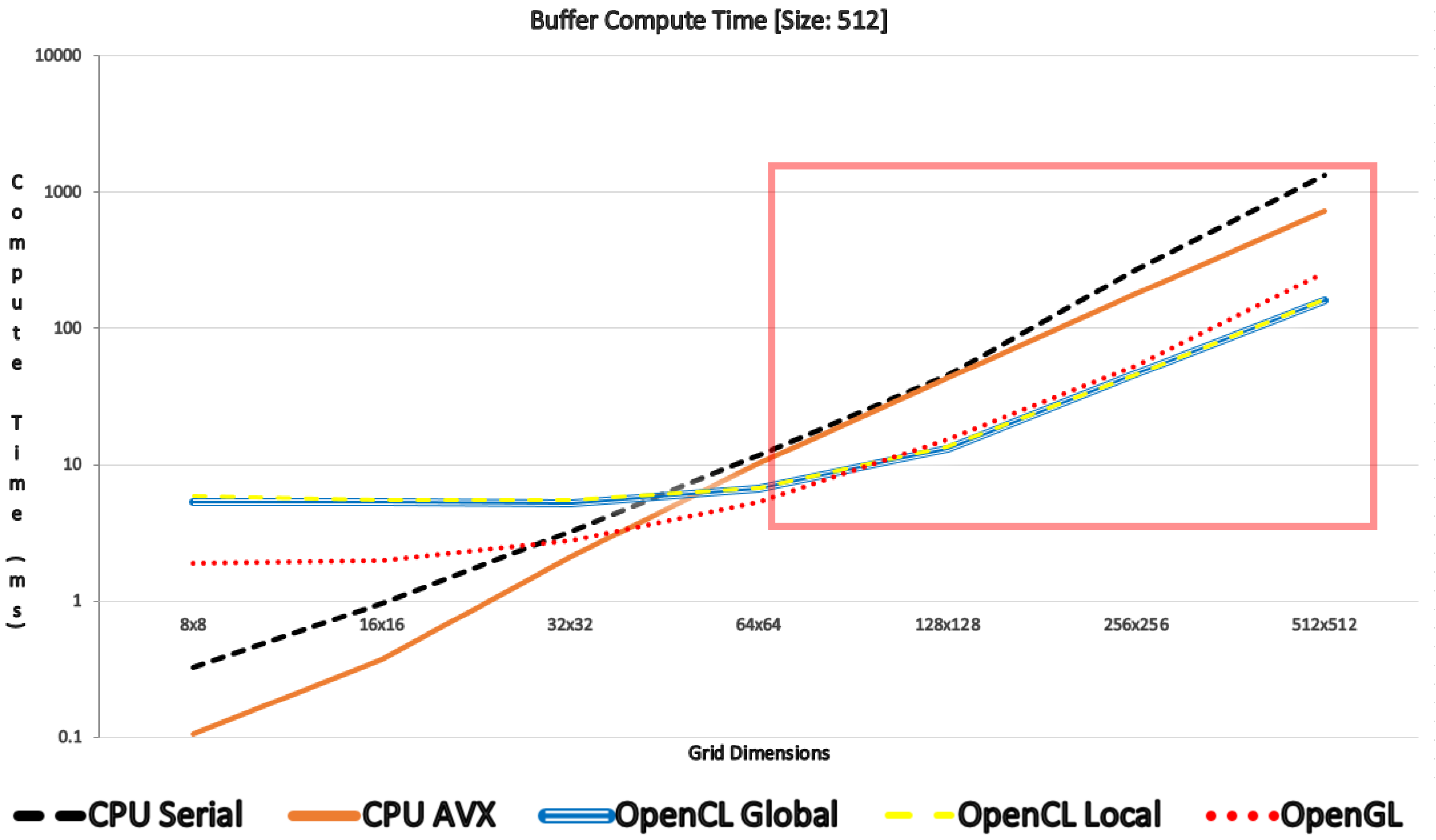
Time per 512 samples.

Results: Model Dimensions

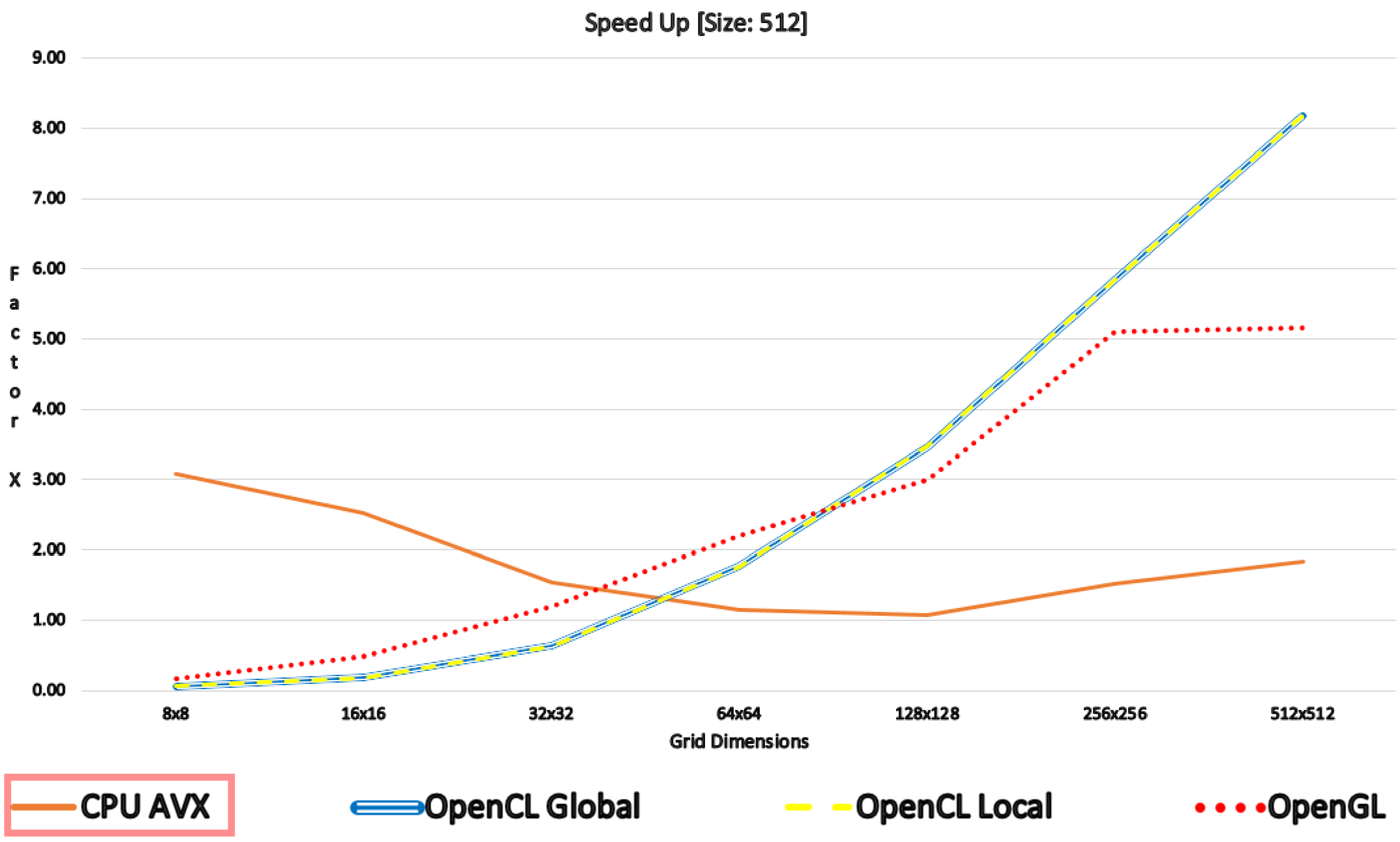
Compute time at smaller dimensions



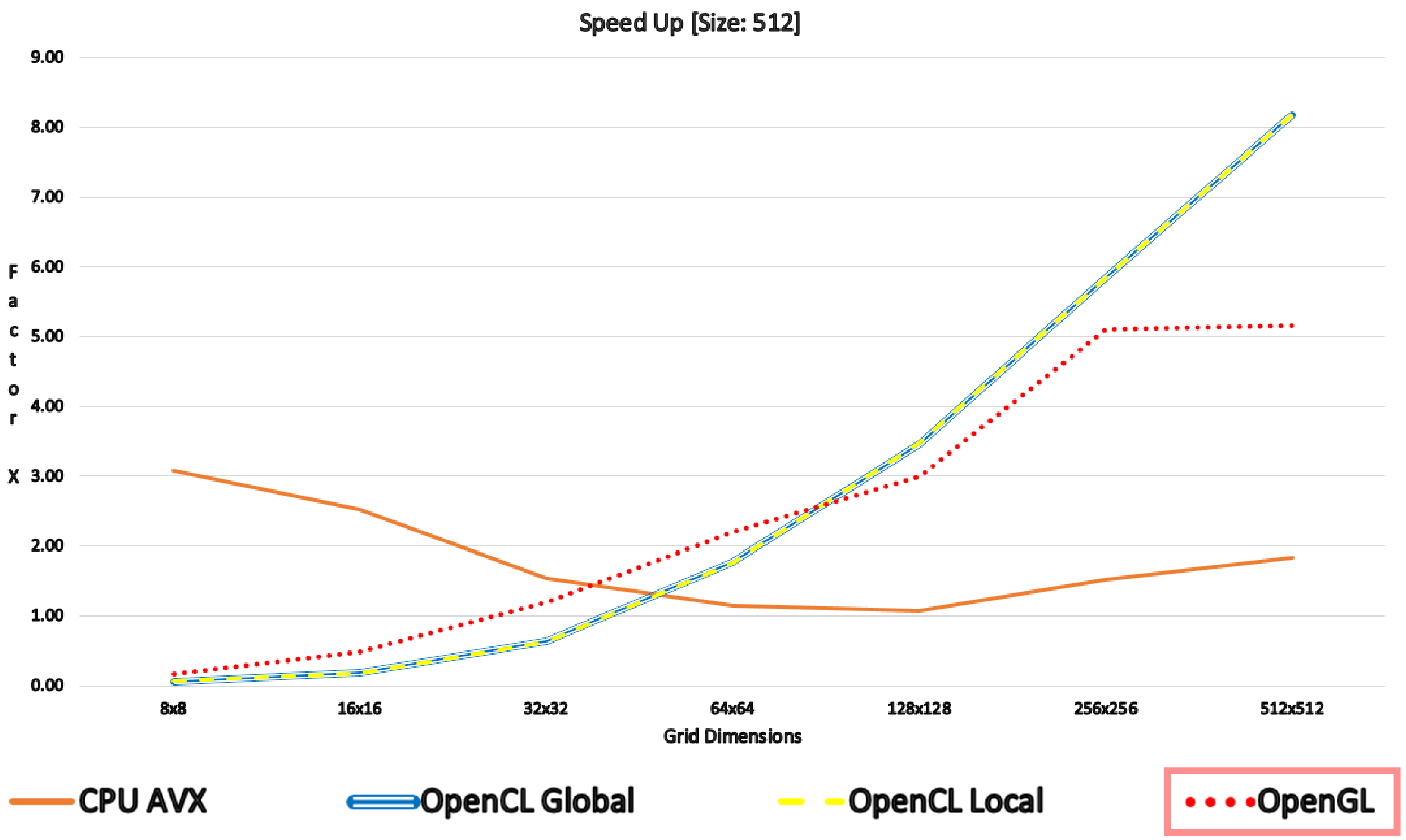
Compute time at higher dimensions



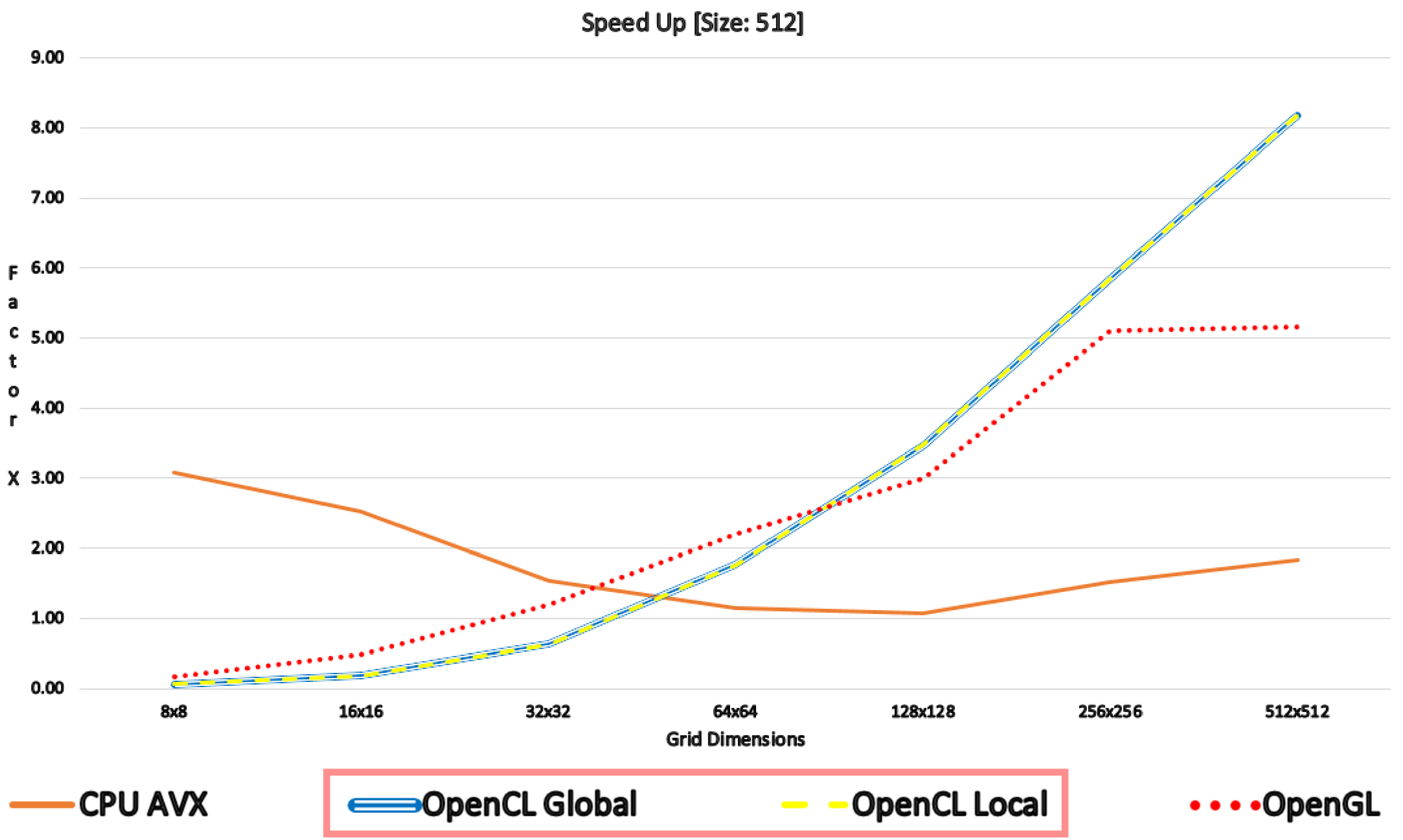
Speedup of AVX



Speedup of OpenGL



Speedup of OpenCL



Conclusion

Project Outcomes

- Designed & developed an OpenCL FDTD Synthesizer.
- Compared different methods of implementing an FDTD synthesizer.
- Analysed performance results and reviewed implementations.

Foundations set for further investigation and development in GPGPU.

Future Work

- Investigating another method for GPGPU, Vulkan. (Sellers et al, 2016)
- Developing a comprehensive benchmarking suite to compare OpenCL and Vulkan.
- Benchmarking on a broader range of systems.

References

[1] Zappi, V., Allen, A. and Fels, S., 2017. Shader-based physical modelling for the design of massive digital musical instruments. In *NIME* (pp. 145-150).
[2] Harris, M., 2012. GPGPU. org. *http://gpgpu. org*.
[3] Sellers, G. and Kessenich, J., 2016. *Vulkan programming guide: The official guide to learning vulkan*. Addison-Wesley Professional.
[4] Maarten Van Walstijn and Konrad Kowalczyk. 2008. On the numerical solution of the 2D wave equation with compact FDTD schemes. Proc. Digital Audio Effects (DAFx), Espoo, Finland (2008), 205–212.