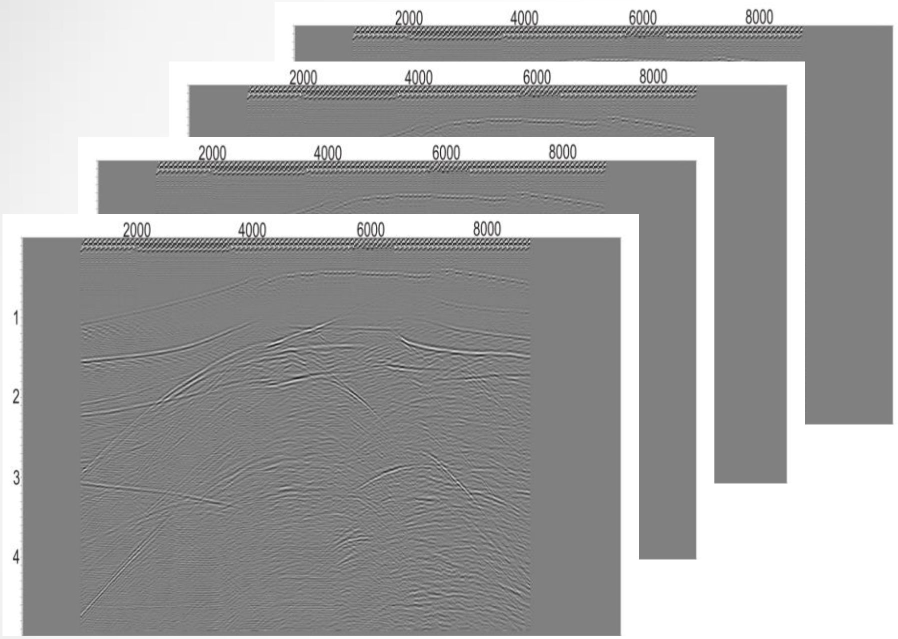
Russian Supercomputing Days

Parallel CPU/GPU Algorithm for One-way Wave Equation Based Migration for Seismic Imaging

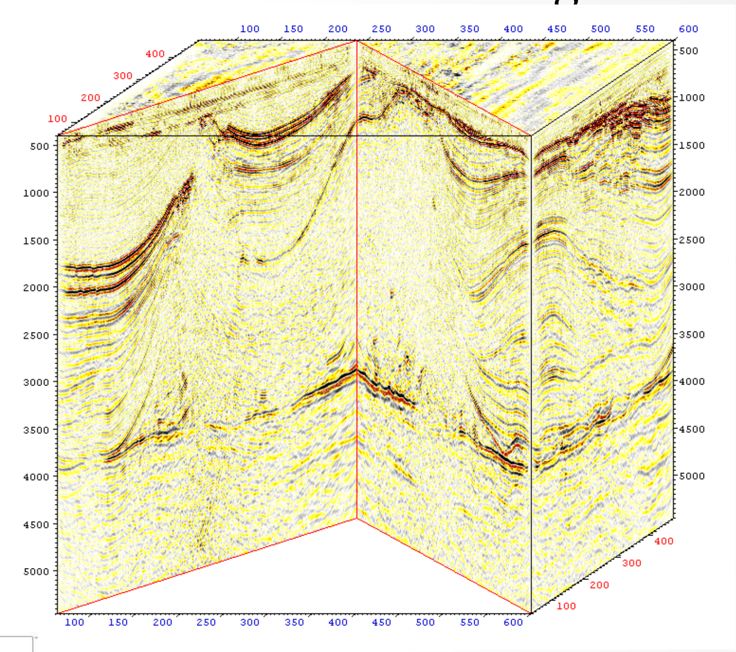
V. Lisitsa, D. Vishnevsky, - IPGG SB RAS
V. Levchenko – IAM RAS
A. Pleshkevich - «CGE»

Statement

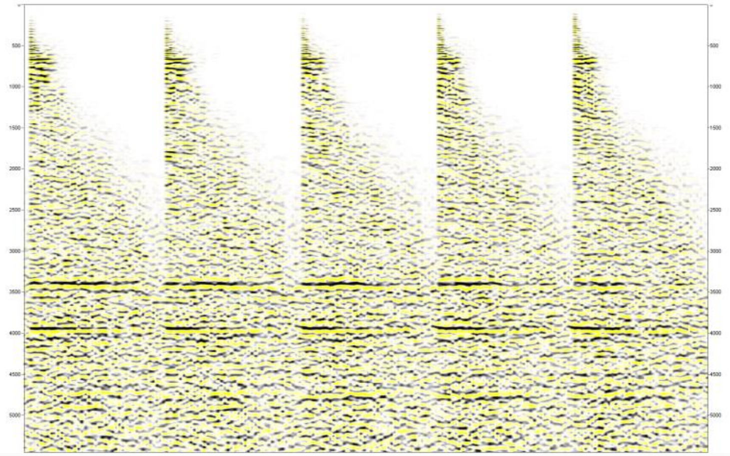
Common-offset data



Common-offset images



CIG



Mathematical background

Imaging condition

$$I(\vec{x}) = \iint G(\vec{x}, \vec{x}_s, \omega) f(\vec{x}_s, \vec{x}_r, \omega) G(\vec{x}, \vec{x}_r, \omega) d\omega d\vec{x}_s d\vec{x}_r$$

Mathematical background

Imaging condition

$$I(\vec{x}) = \iint G(\vec{x}, \vec{x}_s, \omega) f(\vec{x}_s, \vec{x}_r, \omega) G(\vec{x}, \vec{x}_r, \omega) d\omega d\vec{x}_s d\vec{x}_r$$

One-way wave equation

$$\left(\frac{\partial}{\partial z} - \sqrt{\frac{\omega^2}{v^2(x,y,z)} - \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2}} \right)[G] = 0, \quad G(\omega, x, y, 0) = \delta(x - x_0, y - y_0)$$

Pseudo-spectral method

$$
\begin{aligned}
u(\omega,x,y,z+\Delta z,v) &= \alpha_0 u(\omega,x,y,z) + \\
&+\alpha_j F^{-1}\left[\exp\left(i\omega\Delta z\sqrt{\frac{1}{v_j^2}-\frac{k_x^2}{\omega^2}-\frac{k_y^2}{\omega^2}}\right)F\left[u(\omega,x,y,z)\right]\right]+ \\
&+\alpha_{j+1} F^{-1}\left[\exp\left(i\omega\Delta z\sqrt{\frac{1}{v_j^2}-\frac{k_x^2}{\omega^2}-\frac{k_y^2}{\omega^2}}\right)\boxed{F\left[u(\omega,x,y,z)\right]}\right],
\end{aligned}
$$

FFT of the solution at current level

with reference velocities $1500 = v_0 < v_1 < \ldots < v_{J-1} < v_J = 7000$

Russian Supercomputing Days

Pseudo-spectral method

$$u(\omega, x, y, z+\Delta z, v) = \alpha_0 u(\omega, x, y, z) +$$

$$+\alpha_j F^{-1}\left[\exp\left(i\omega\Delta z\sqrt{\frac{1}{v_j^2}-\frac{k_x^2}{\omega^2}-\frac{k_y^2}{\omega^2}}\right)F\left[u(\omega, x, y, z)\right]\right]+$$

$$+\alpha_{j+1} F^{-1}\left[\boxed{\exp\left(i\omega\Delta z\sqrt{\frac{1}{v_j^2}-\frac{k_x^2}{\omega^2}-\frac{k_y^2}{\omega^2}}\right)\boxed{F\left[u(\omega, x, y, z)\right]}}\right],$$

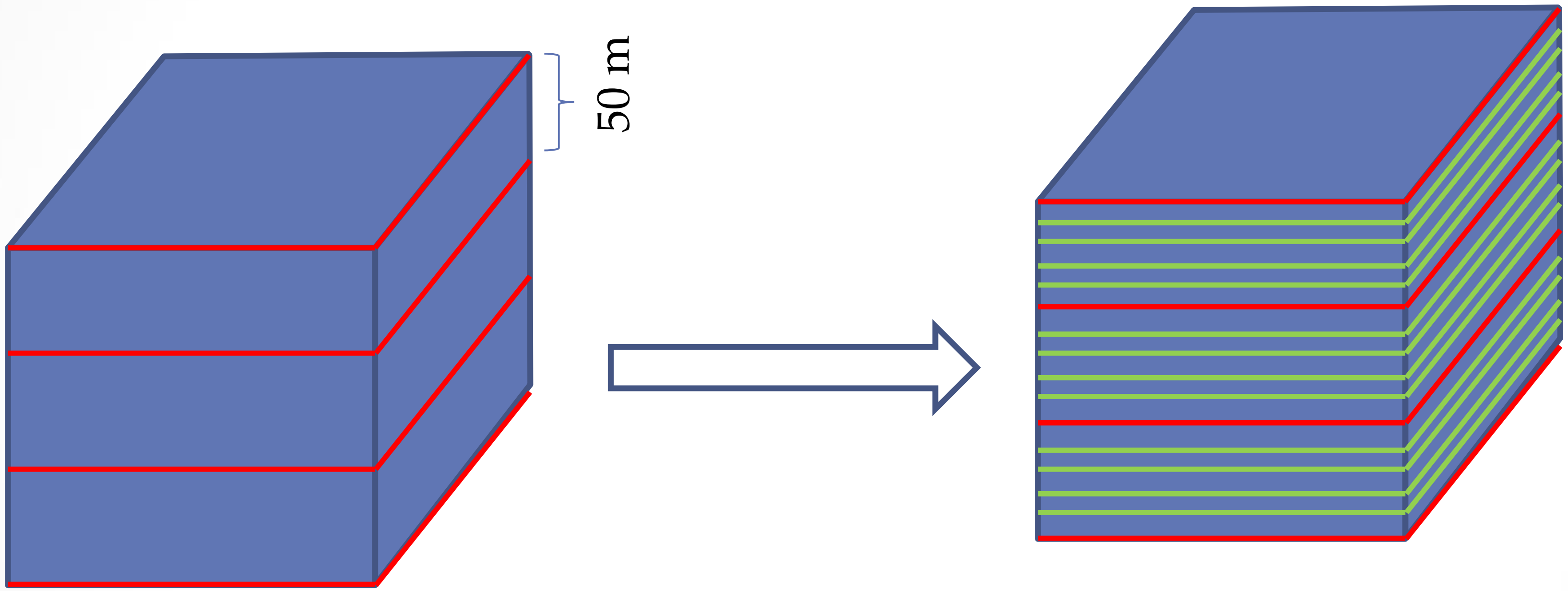
One-way wave operator action

Pseudo-spectral method

$$
\begin{aligned}
u(\omega, x, y, z+\Delta z, v) &= \alpha_0 u(\omega, x, y, z) + \\
&+\alpha_j F^{-1}\left[\exp\left(i\omega\Delta z\sqrt{\frac{1}{v_j^2}-\frac{k_x^2}{\omega^2}-\frac{k_y^2}{\omega^2}}\right)F\left[u(\omega, x, y, z)\right]\right] + \\
&+\alpha_{j+1}\boxed{F^{-1}\left[\boxed{\exp\left(i\omega\Delta z\sqrt{\frac{1}{v_j^2}-\frac{k_x^2}{\omega^2}-\frac{k_y^2}{\omega^2}}\right)\boxed{F\left[u(\omega, x, y, z)\right]}}\right]},
\end{aligned}
$$

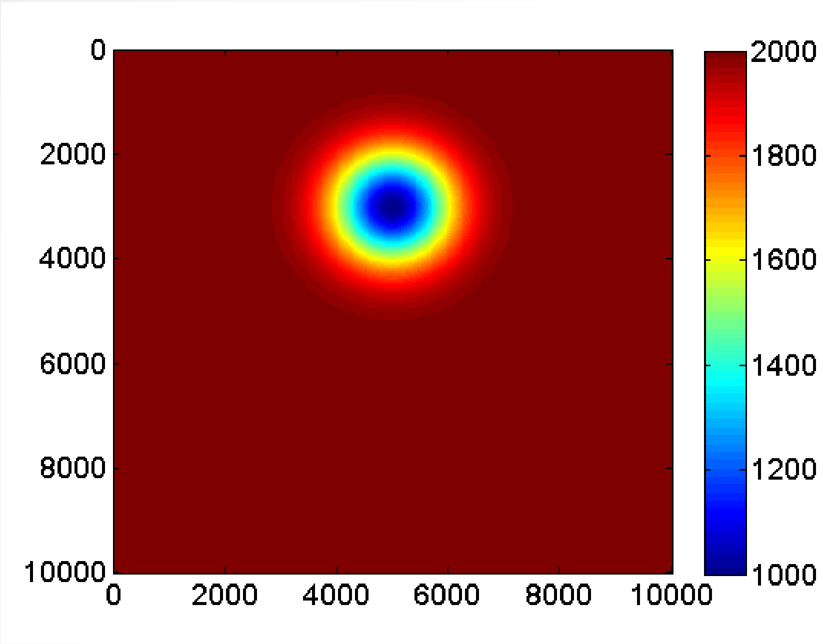
iFFT –to compute solution at z+dz

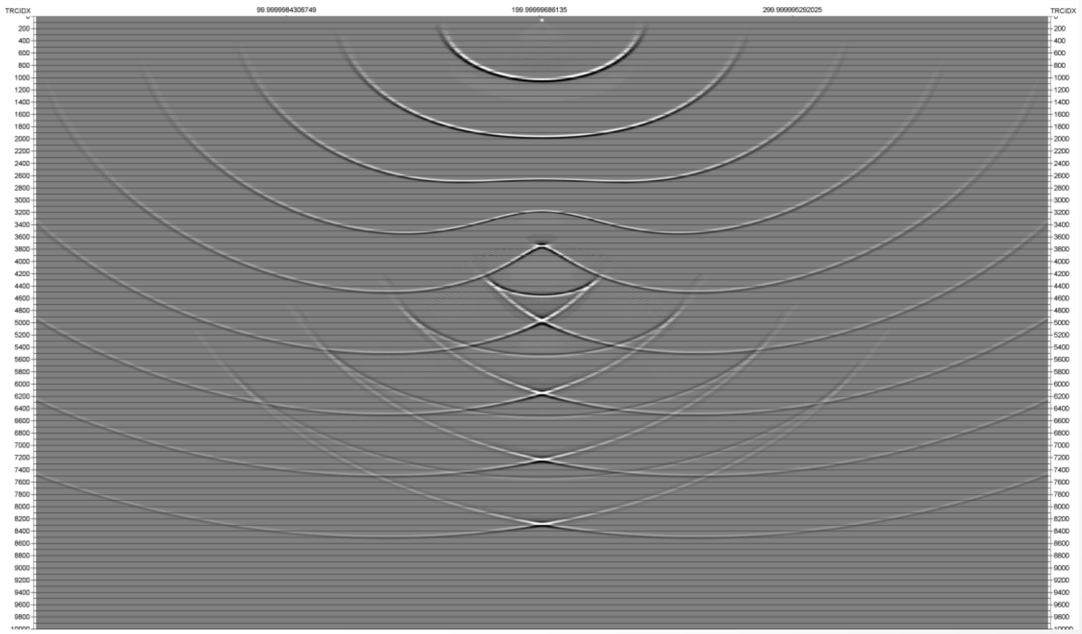
Interpolation



$$u(\omega,x,y,\tilde{z})=\frac{z+\Delta z-\tilde{z}}{\Delta z}u(\omega,x,y,z)\exp\left(-i\frac{\omega}{v}(\tilde{z}-z)\right)+\frac{\tilde{z}-z}{\Delta z}u(\omega,x,y,z+\Delta z)\exp\left(-i\frac{\omega}{v}(\tilde{z}-z-\Delta z)\right)$$

Examples of the wavefields



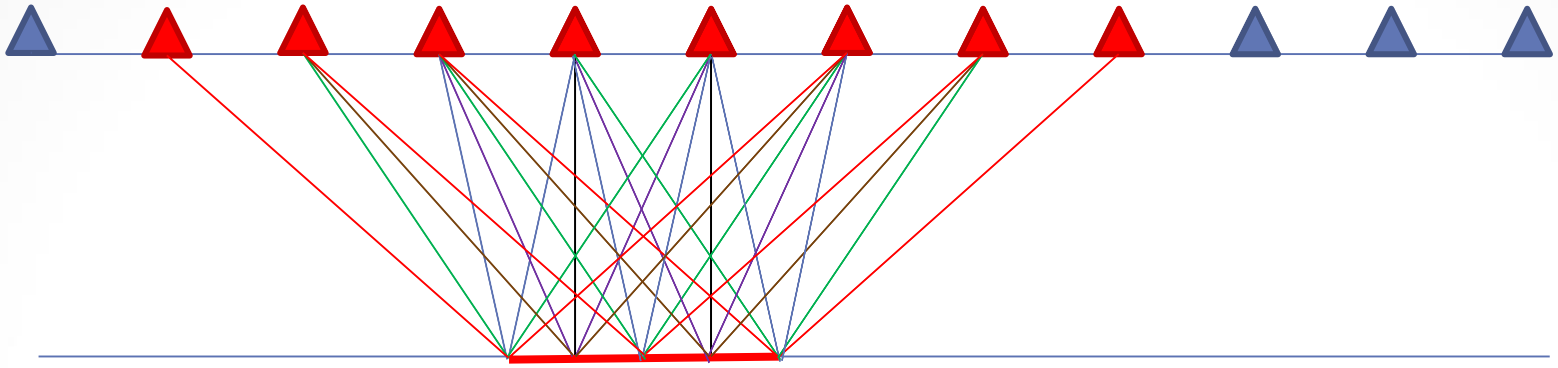


Russian Supercomputing Days

The algorithm

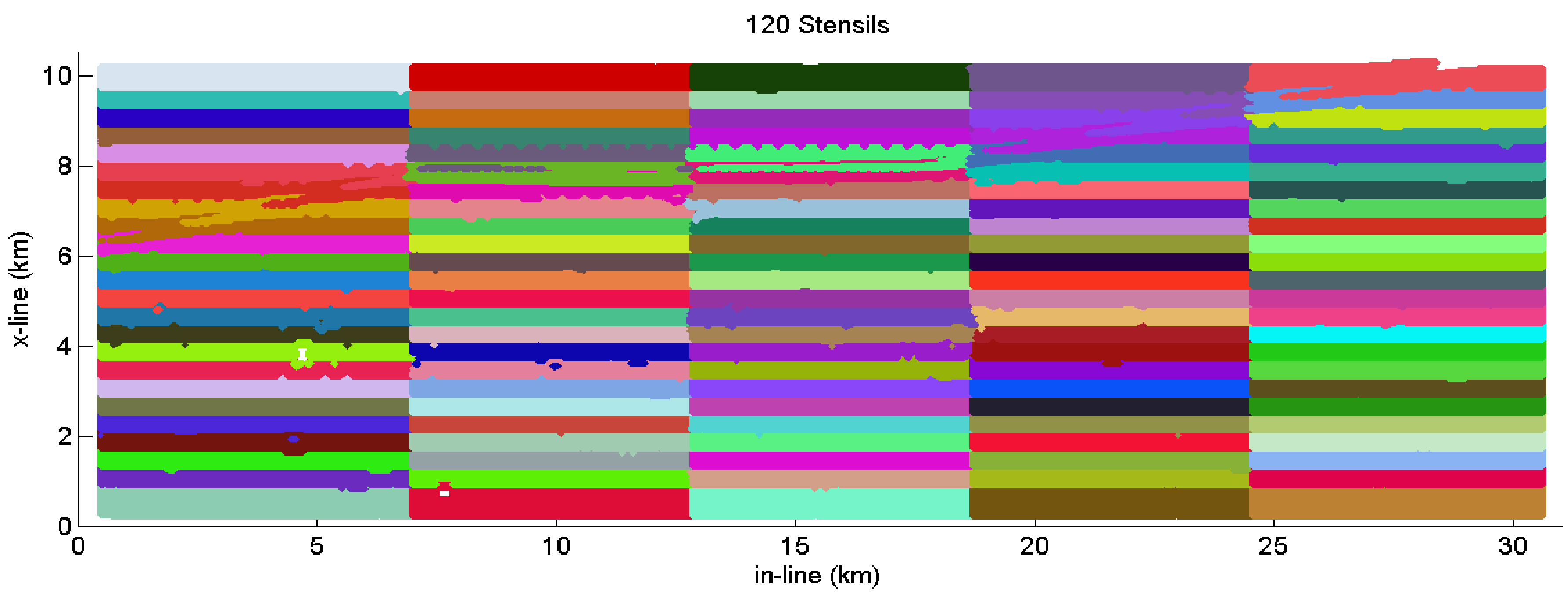
1. We need the common-offset common-azimuth images;
2. Computation of the Green’s function is the most time consuming part;
3. We can compute solution and image layer-by-layer, thus dealing with 2D problems;

Parallelism wrt data



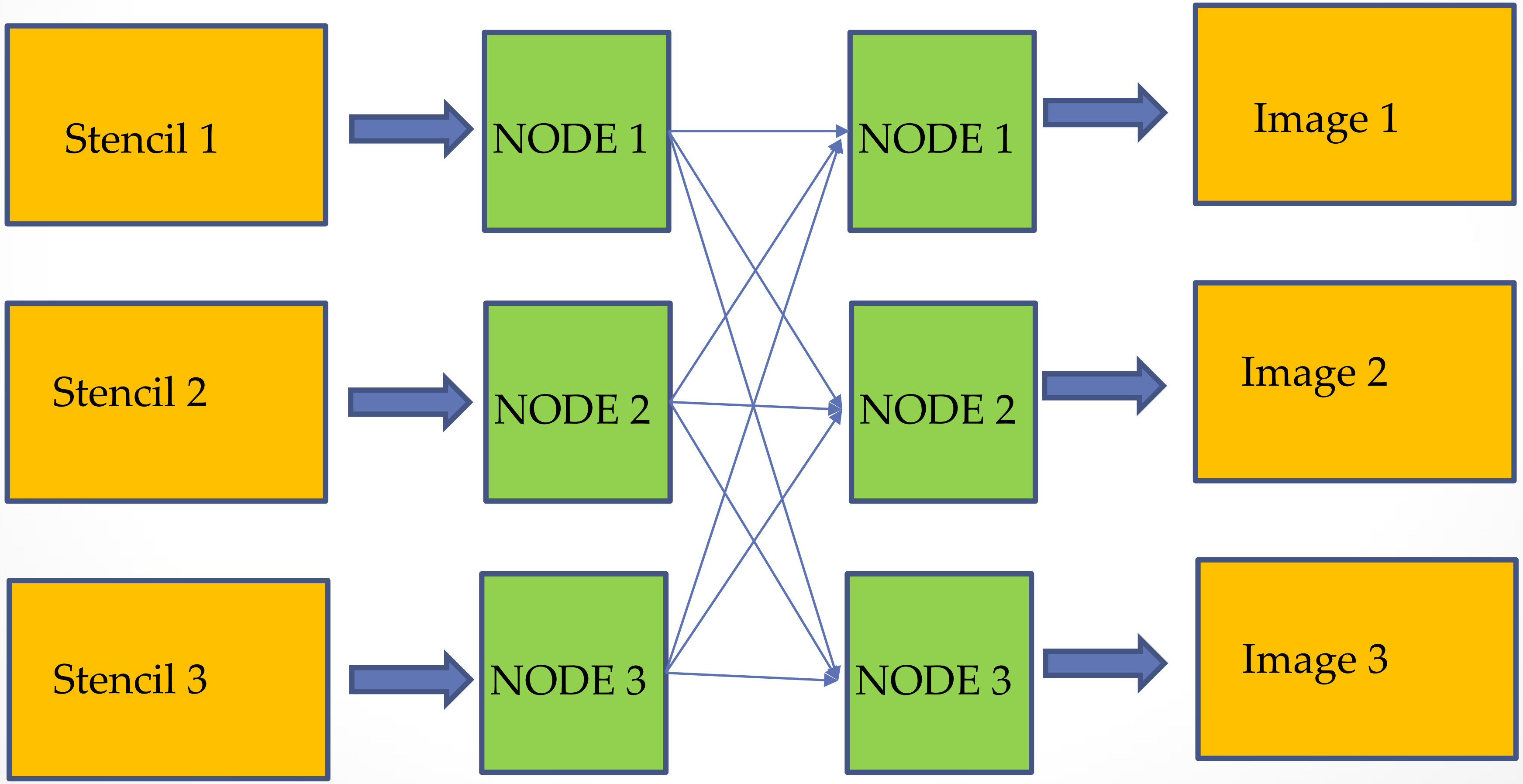
Stencil is a set of common mid-points with all the sources and receivers, adjusted to these mid-points, traces, etc.

Parallelism wrt data

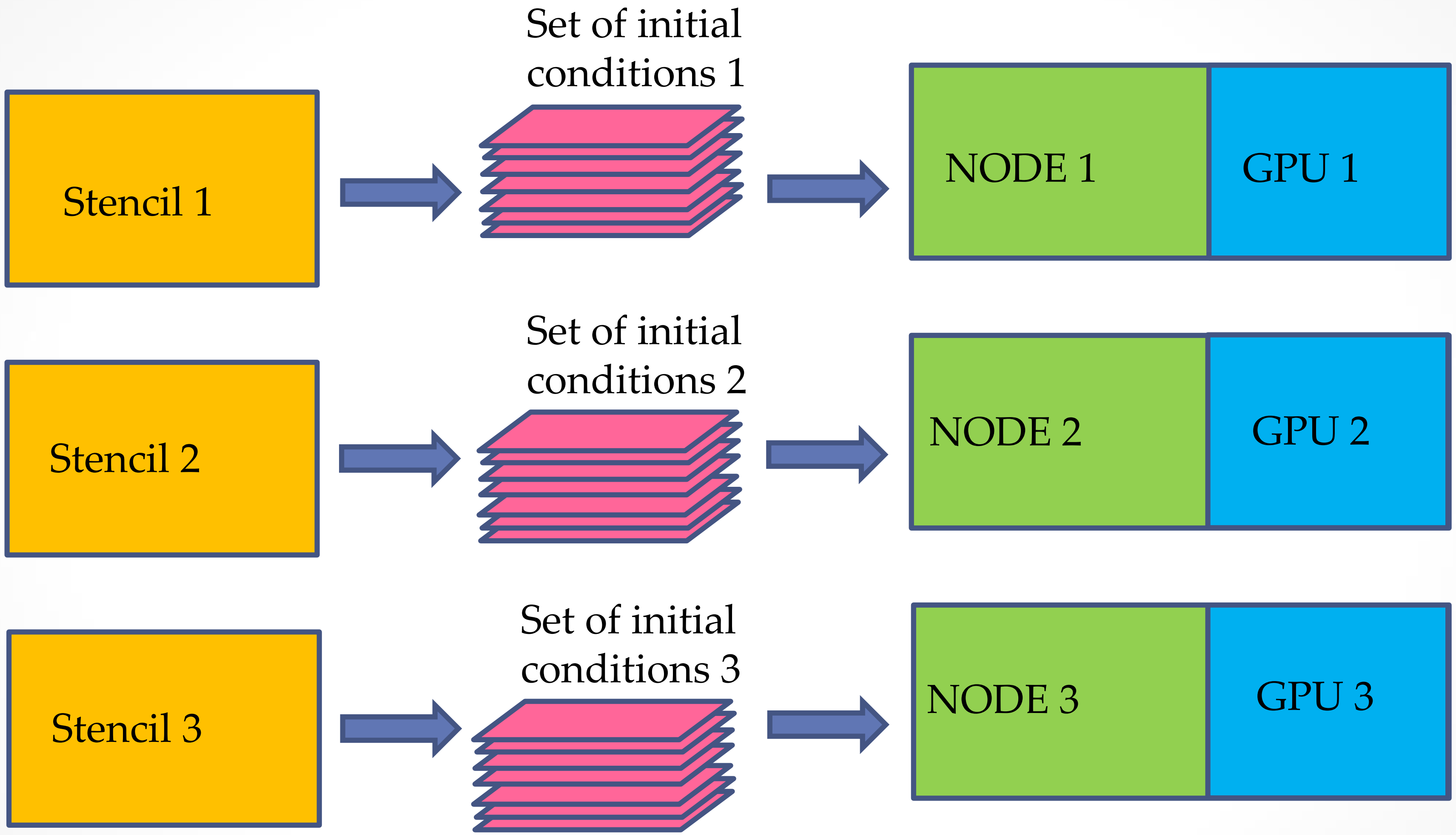


Algorithm

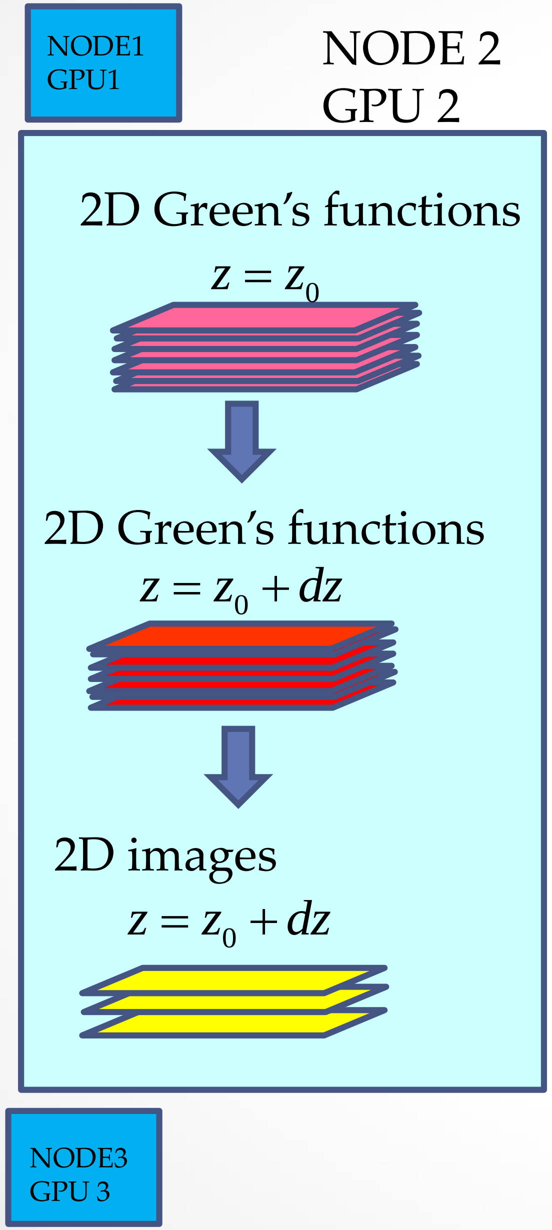
1. Input data: 1 node – 1 stencil
2. Output data: 1 node – 1 image



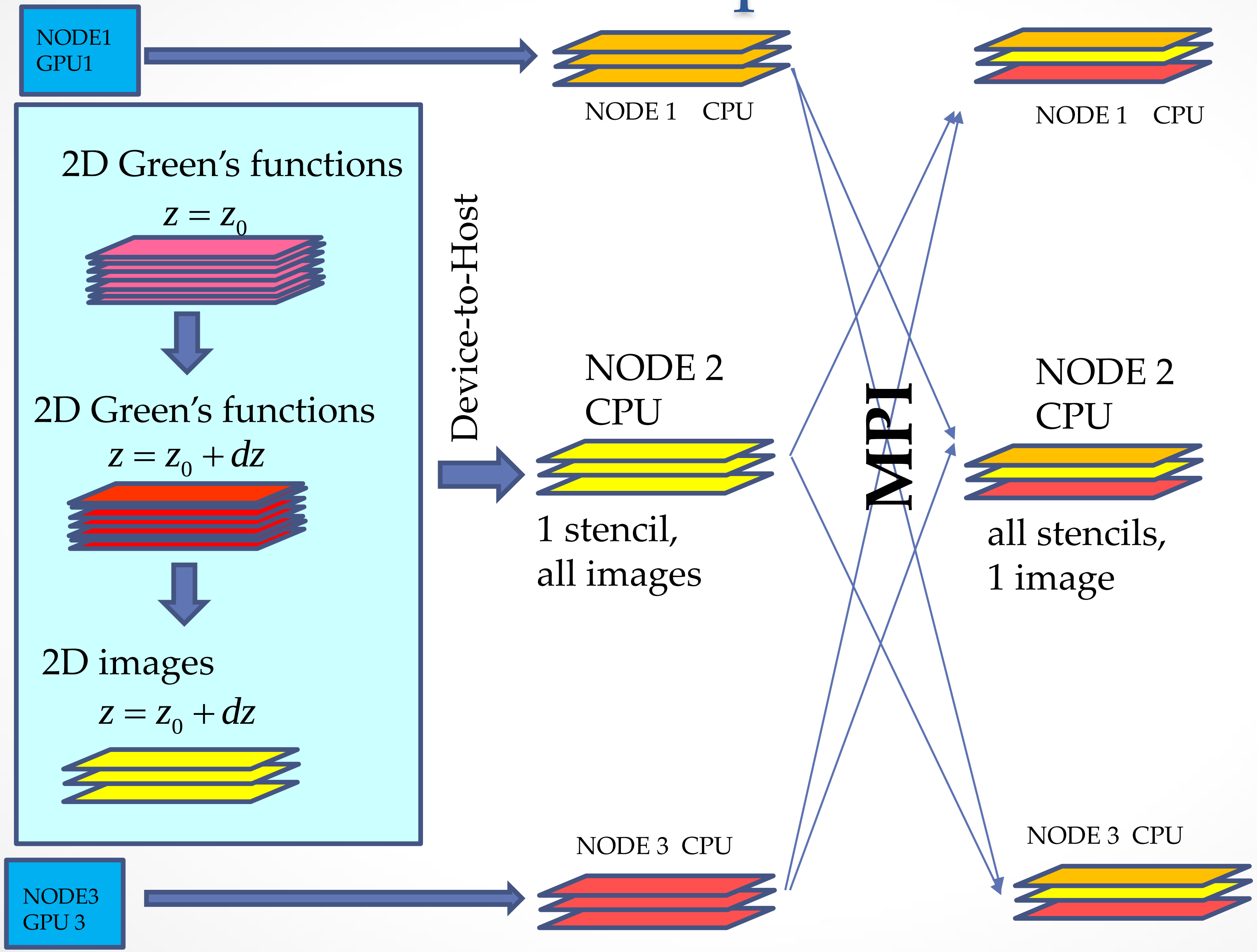
Algorithm


Set of initial conditions 1
Stencil 1
NODE 1
GPU 1
Set of initial conditions 2
Stencil 2
NODE 2
GPU 2
Set of initial conditions 3
Stencil 3
NODE 3
GPU 3

Step 1

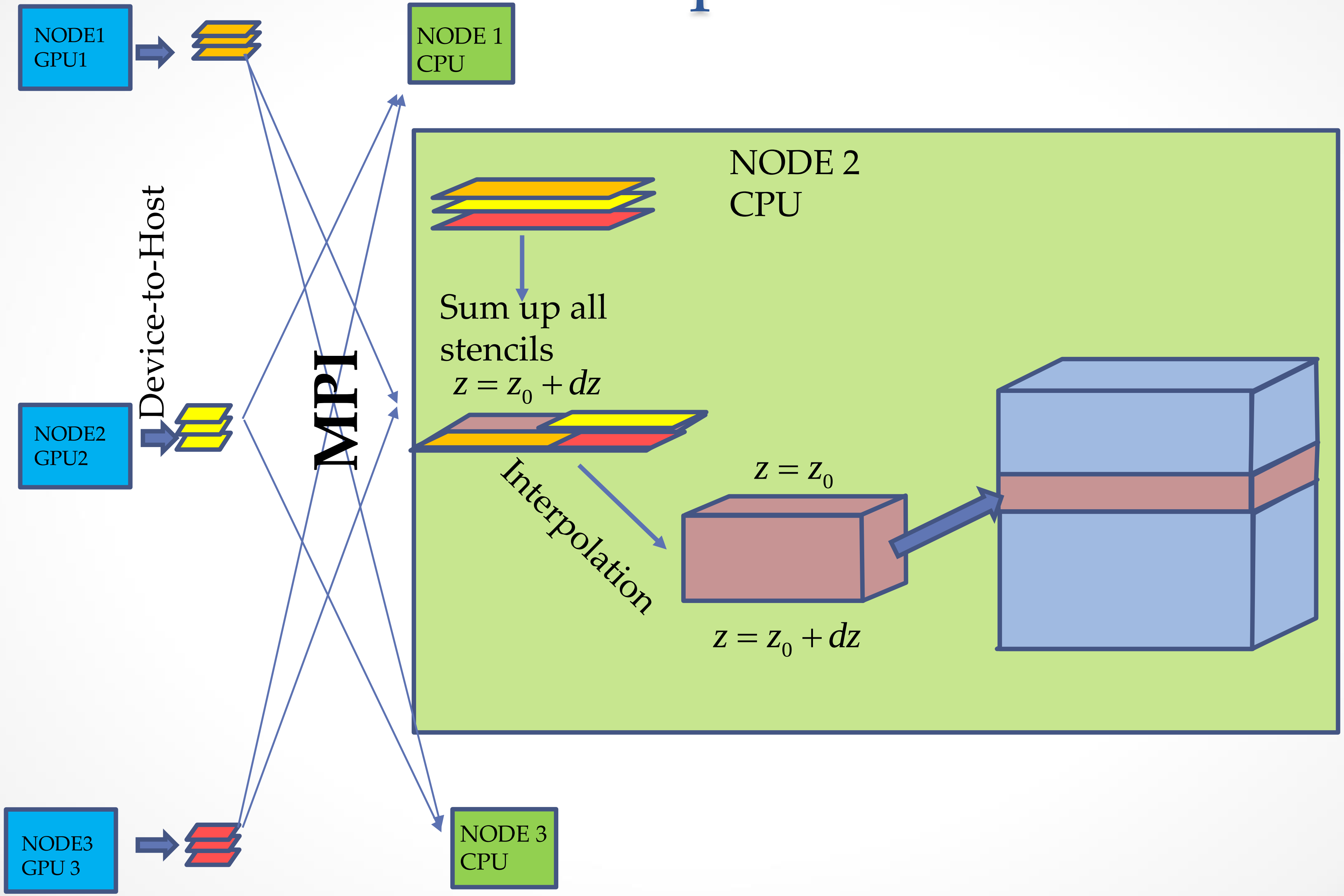

NODE1
GPU1
NODE 2
GPU 2
2D Green's functions
$z = z_0$
2D Green's functions
$z = z_0 + dz$
2D images
$z = z_0 + dz$
NODE3
GPU 3

Step 2

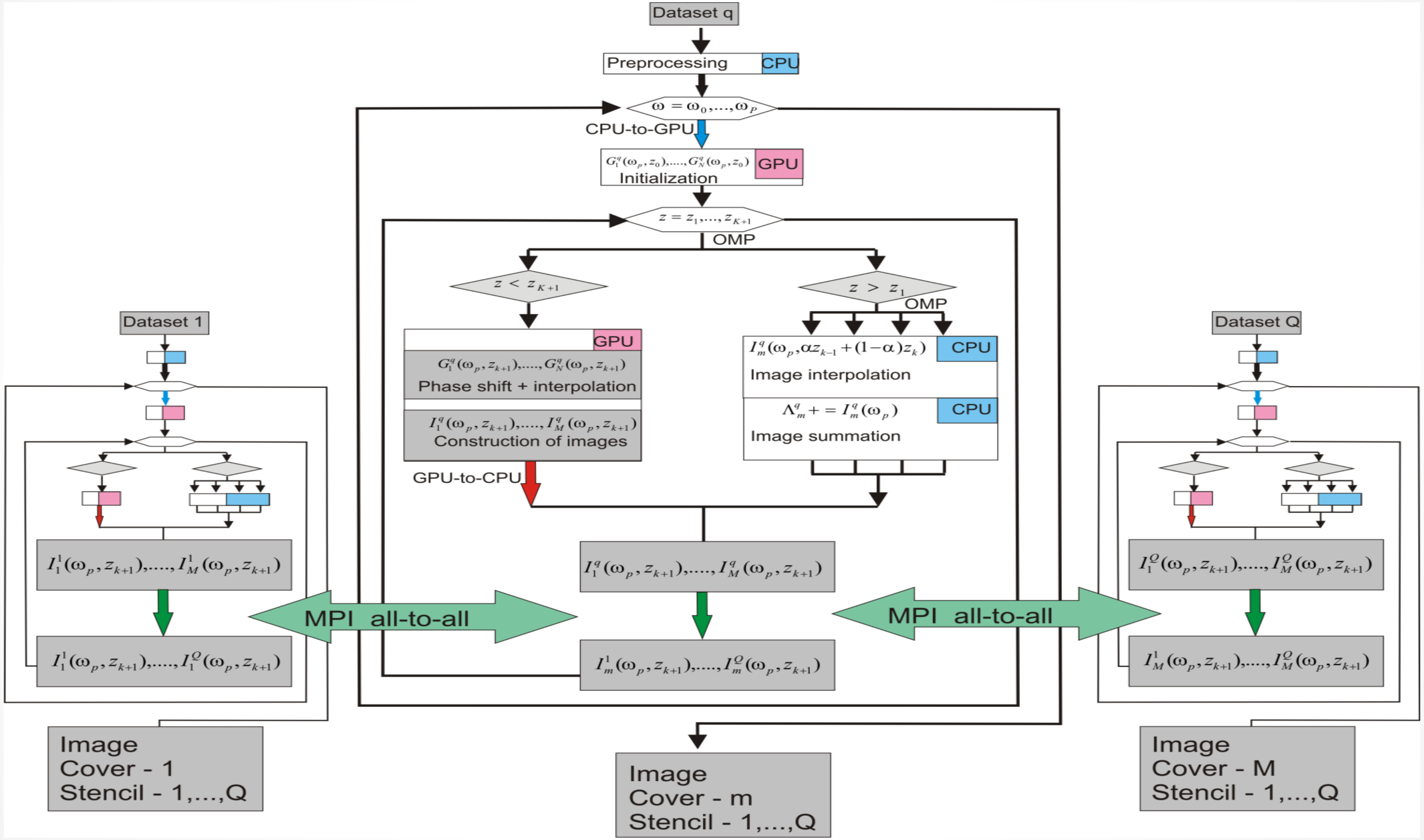

NODE1
GPU1
2D Green's functions
$z = z_0$
2D Green's functions
$z = z_0 + dz$
2D images
$z = z_0 + dz$
NODE3
GPU 3
Device-to-Host
NODE 1 CPU
NODE 2
CPU
1 stencil,
all images
NODE 3 CPU
MPI
NODE 1 CPU
NODE 2
CPU
all stencils,
1 image
NODE 3 CPU

Russian Supercomputing Days

Step 3

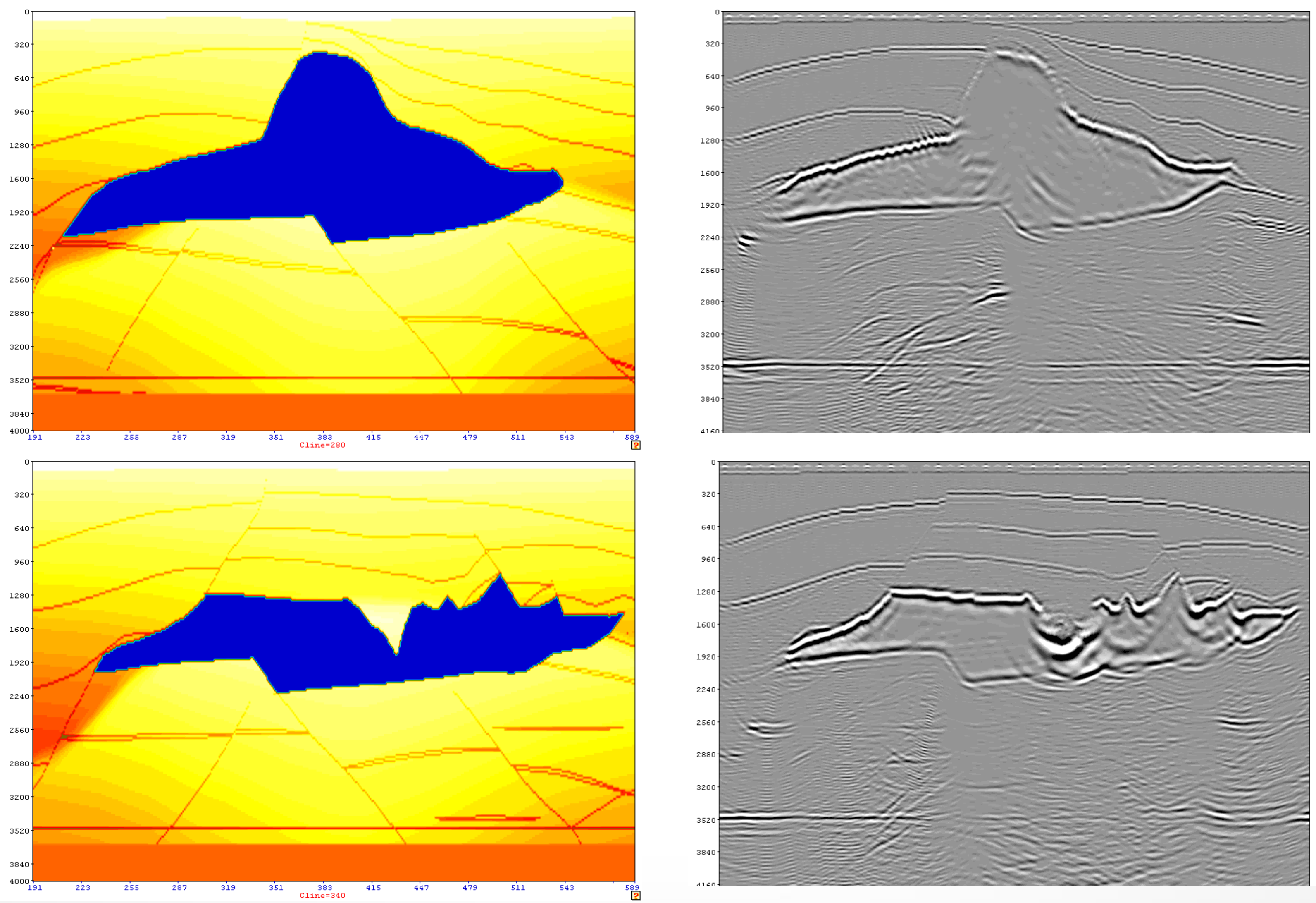


Block-scheme



SEG Salt

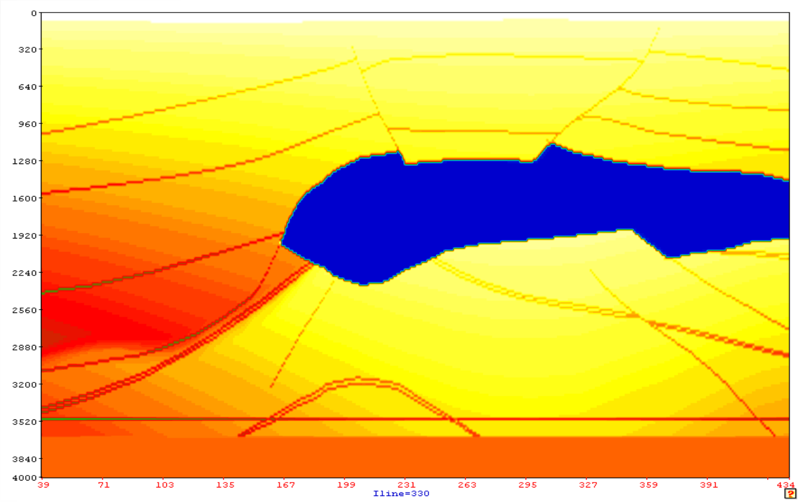
cross-line

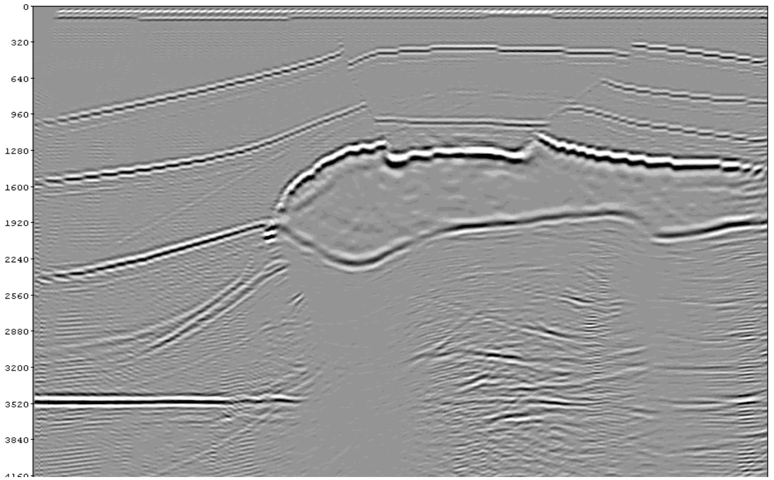


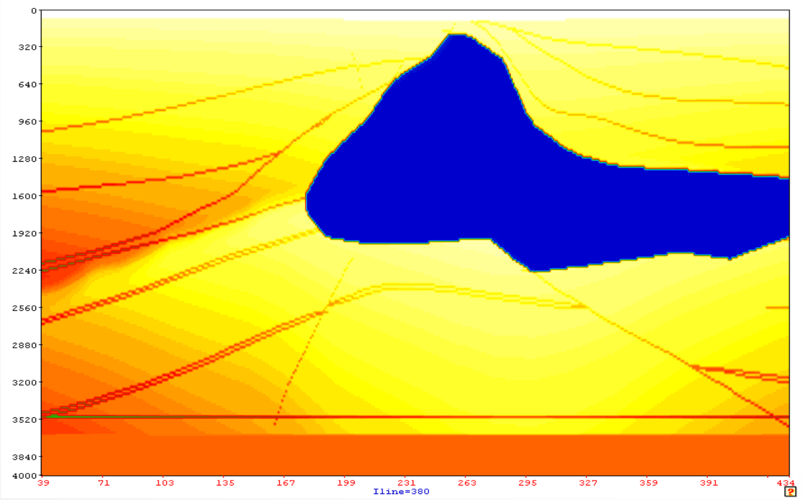
Russian Supercomputing Days

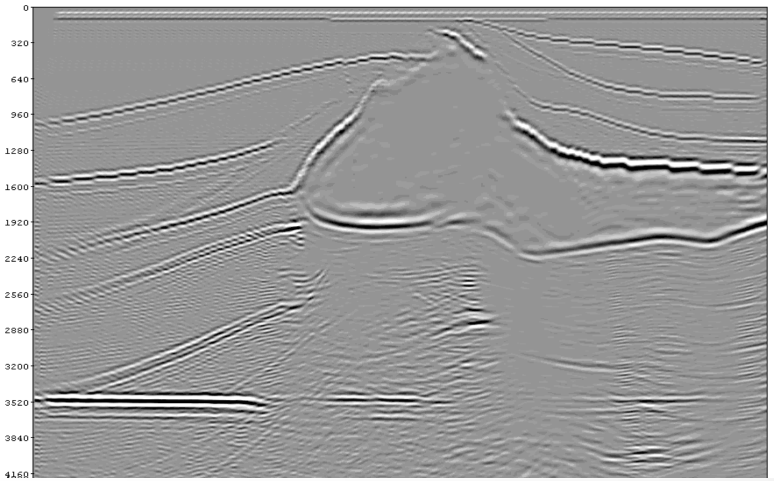
SEG Salt

inline-line

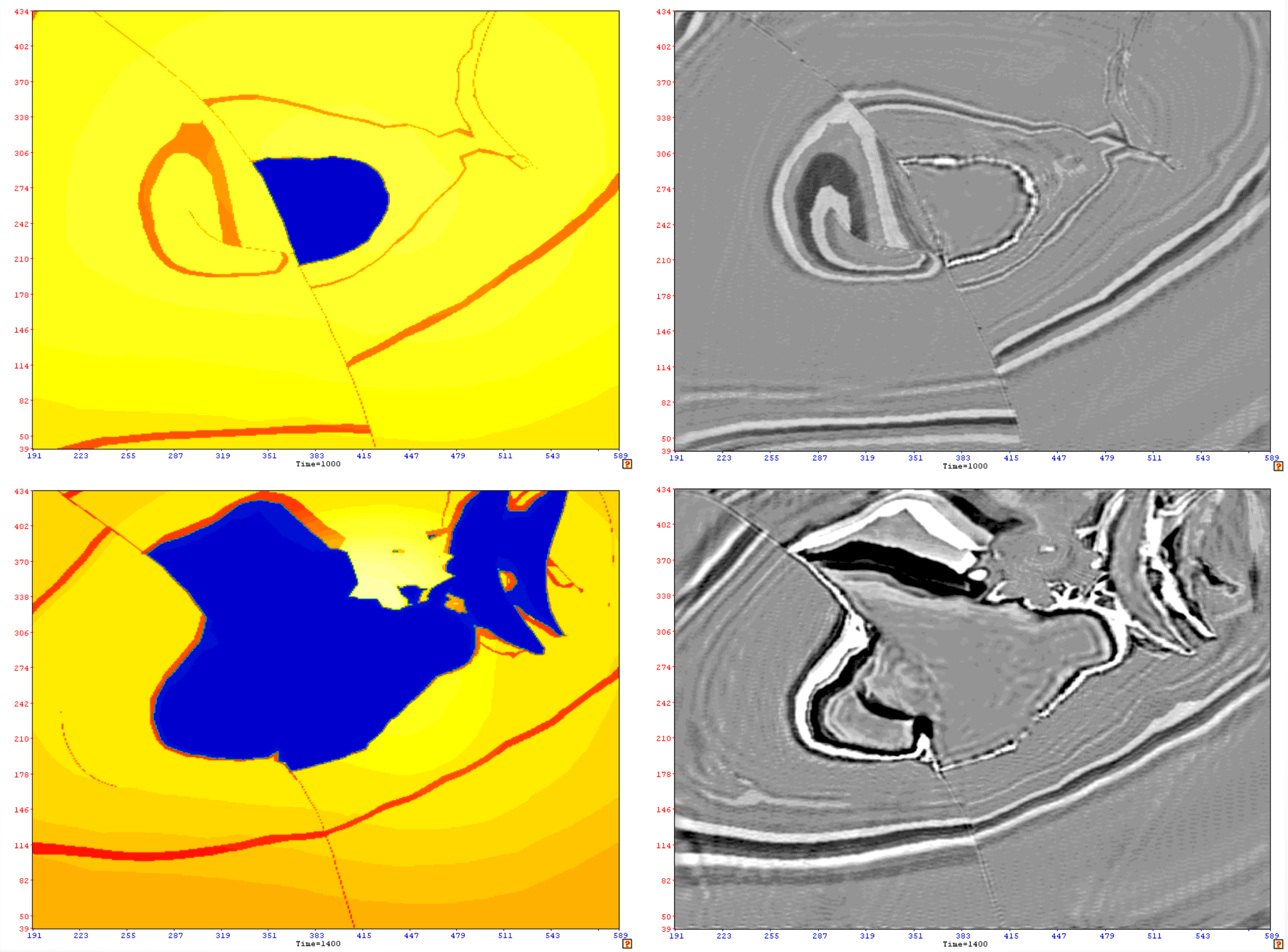




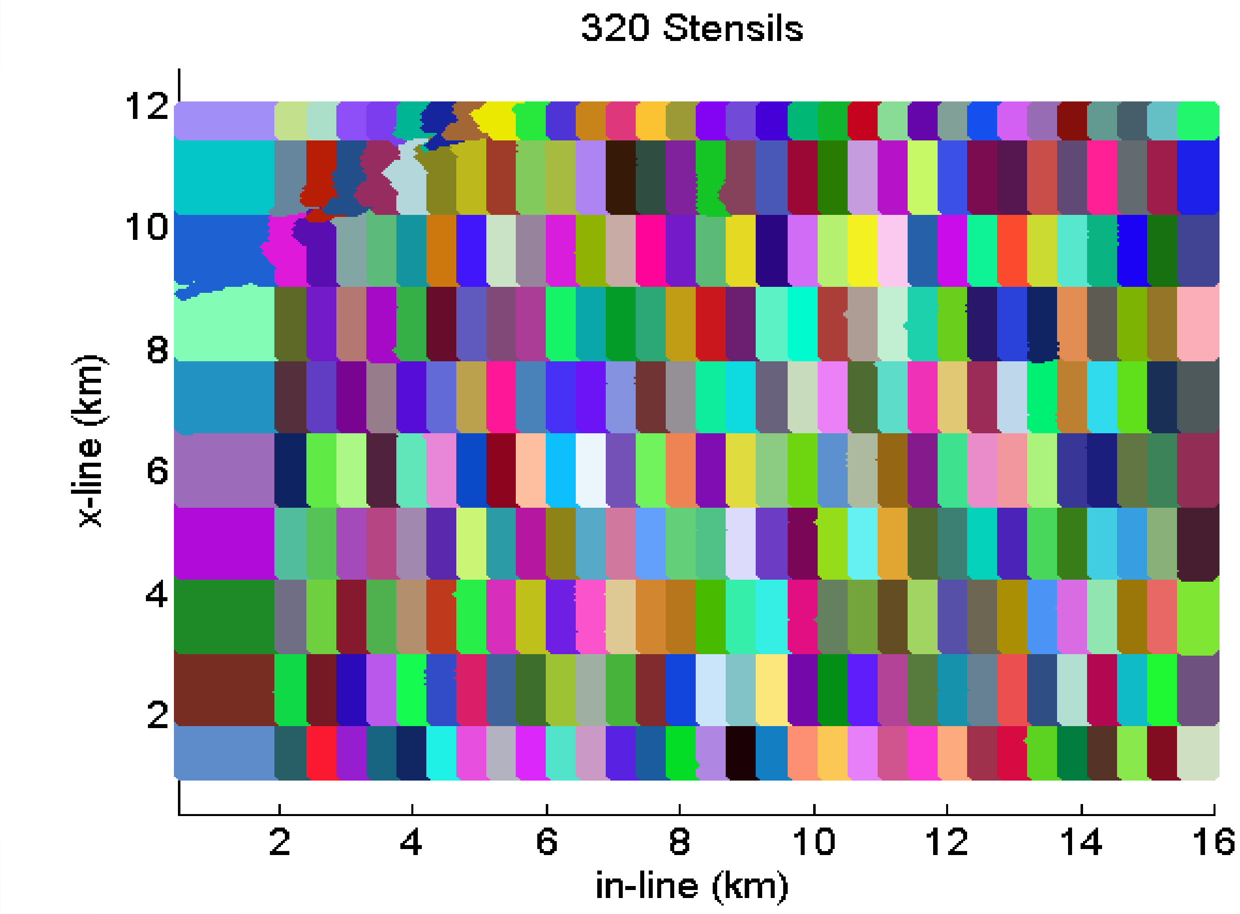




SEG Salt



Field data



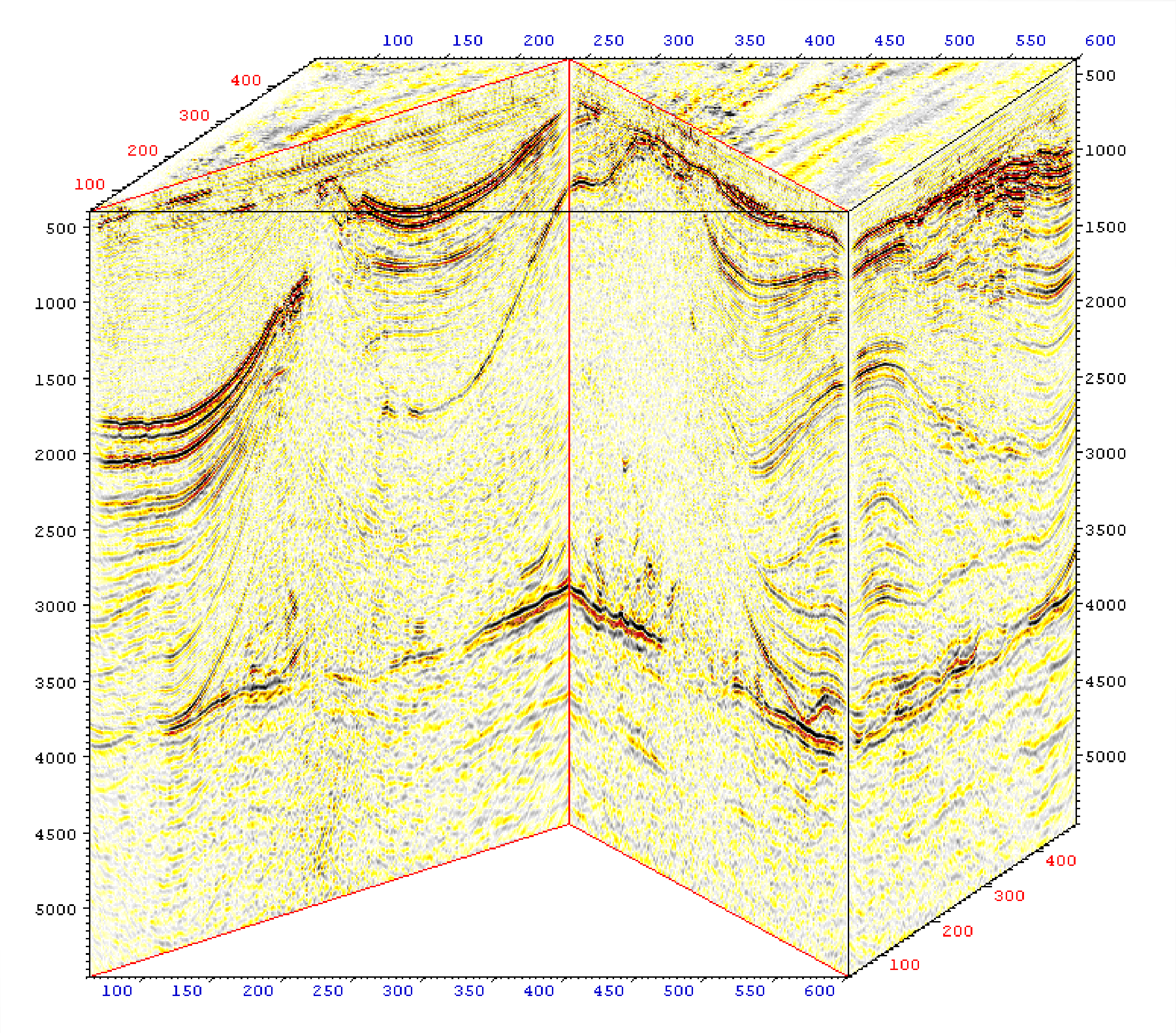
Field data

Common-offset gathers – 80

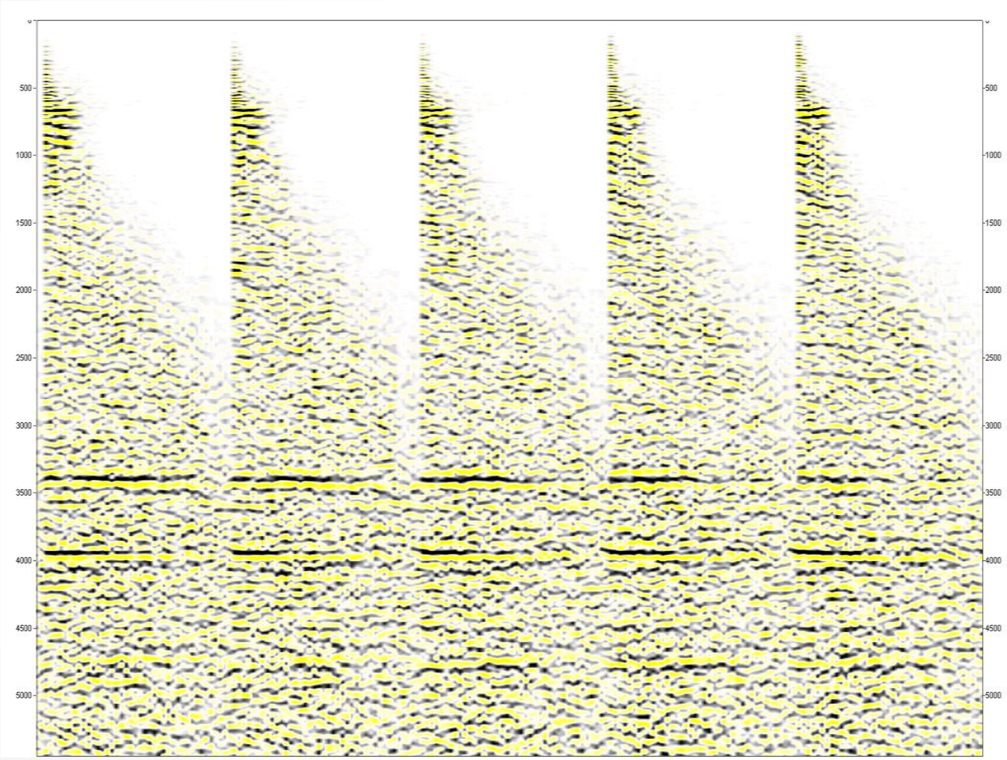
Frequency range 2 – 80 Hz

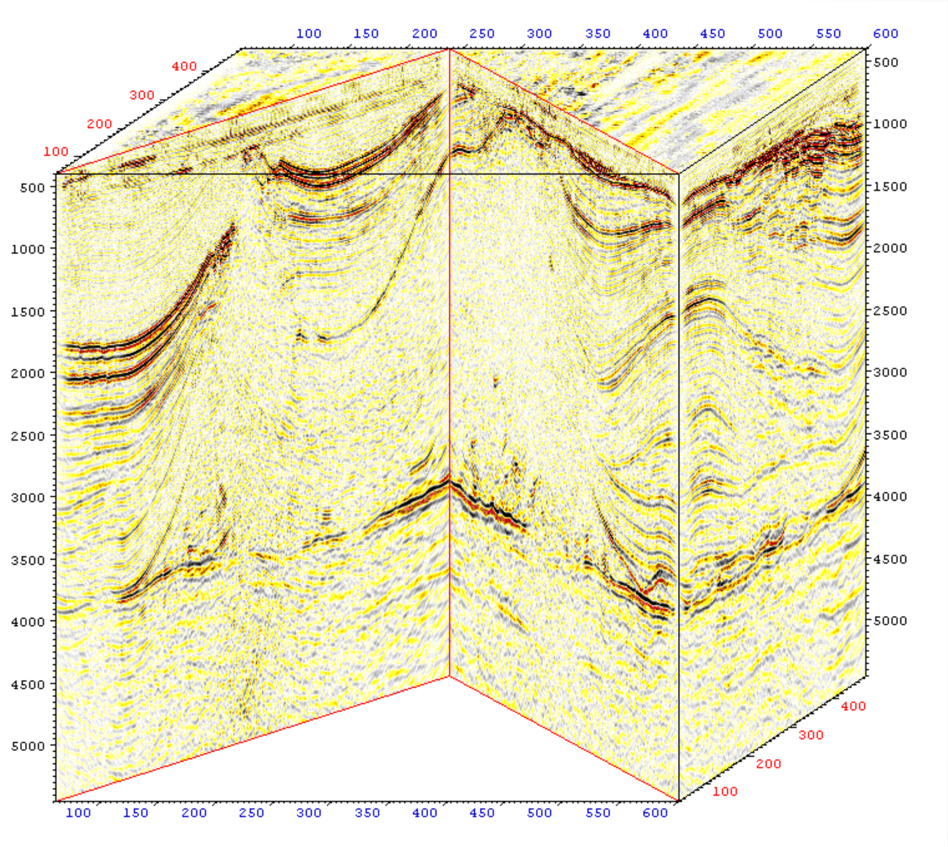
«Lomonosov -2» supercomputer was used

Time – 12000 node-hours.



Real data




100 150 200 250 300 350 400 450 500 550 600
500 1000 1500 2000 2500 3000 3500 4000 4500 5000
100 200 300 400

Conclusions

- We presented highly parallel algorithm for seismic imaging.
- The algorithm is based on the one-way wave migration
- We implemented main computations using CUDA and qFFT
- Dataflow if parallelized by MPI
- Single dataset (stencil) is input per node – single common-offset image is output per node.