GARNET

Thursday Update

Casper Pranger, Marie Bocher
William Sawyer, Andreas Herten

```
////////////////////////////////////////
//                                    //
//   POINT-WISE FUNCTION DEFINITIONS  //
//                                    //
////////////////////////////////////////

auto porosity_evolution = [&]( double P_t, double P_f )
    { return ( P_f - P_t ) / η_φ; };

auto total_density = [&]( double& φc )
    { return ρ_s * ( 1 - φc ) + ρ_f * φc; };

auto bulk_constitutive = [&]( double ϵ )
    { return 2 * η_s * ϵ; };

auto bulk_momentum_balance = [&]( double Δτ, double ΔP_t, double ρ_t, double g )
    { return Δτ - ΔP_t + ρ_t * g; };

auto bulk_mass_balance = [&]( double Δv_s, double P_t, double P_f, double φ )
    { return Δv_s + ( P_t - P_f ) / ( 1 - φ ) / η_φ ); };

auto darcy_flux = [&]( double ΔP_f, double φc, double g )
    { return  -k0 / η_f * std::pow(φc/φ0,3) * ( ΔP_f - ρ_f * g ); };

auto fluid_mass_momentum_balance = [&]( double Δq_D, double P_t, double P_f, double φ )
    { return Δq_D - ( P_t - P_f ) / ( 1 - φ ) / η_φ ); };
```

__host__ __device__

GARNET

```
////////////////////////////////////////
//                                    //
//   OBJECTIVE FUNCTION DEFINITION    //
//                                    //
////////////////////////////////////////

auto residual_evaluation = [&]( auto& Rv_s, auto& RP_t, auto& RP_f )
{
    // Porosity evolution
    φ.SetAlpha( 0. );
    φ.SetBeta ( porosity_evolution, P_t[0], P_f[0] );
    φ.TrivialSolve<BDF<1>>();

    φc.IsInterpolationOf( φ[0] );

    // Bulk momentum balance
    τ   .Set( bulk_constitutive, ϵ().RemoveTrace() );
    ρ_t .Set( total_density, φc );
    Rv_s.Set( bulk_momentum_balance, Δτ(), ΔP_t(), ρ_t, g );

    // Bulk mass balance
    RP_t.Set( bulk_mass_balance, Δv_s(), P_t[0], P_f[0], φ[0] );

    // Fluid mass and momentum balance
    q_D .Set( darcy_flux, ΔP_f(), φc, g );
    RP_f.Set( fluid_mass_momentum_balance, Δq_D(), P_t[0], P_f[0], φ[0] );
};
```

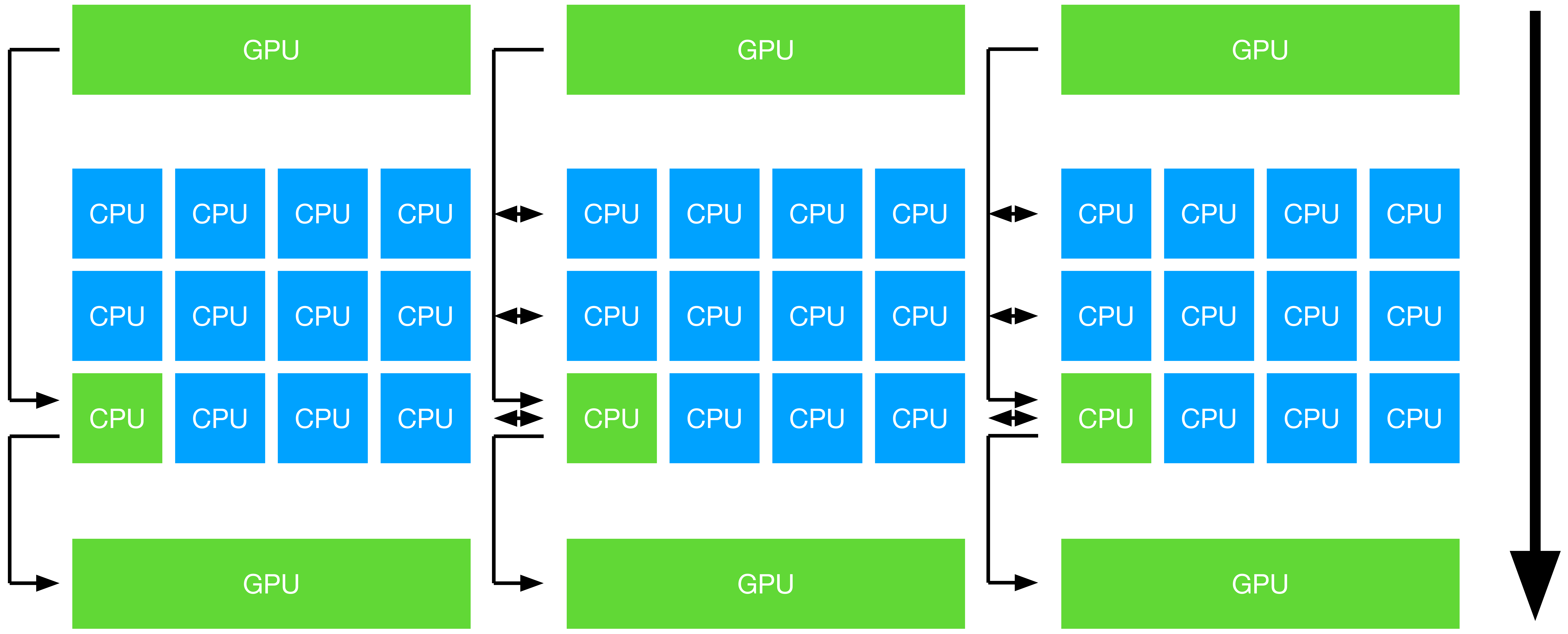
Working on GPU

GARNET

```
//////////////////////////////////////
//                                  //
//   OBJECTIVE FUNCTION DEFINITION  //
//                                  //
//////////////////////////////////////

auto residual_evaluation = [&]( auto& Rv_s, auto& RP_t, auto& RP_f )
{
    // Porosity evolution
    φ.SetAlpha( 0. );
    φ.SetBeta ( porosity_evolution, P_t[0], P_f[0] );
    φ.TrivialSolve<BDF<1>>();

    φc.IsInterpolationOf( φ[0] );

    // Bulk momentum balance
    τ   .Set( bulk_constitutive, ϵ().RemoveTrace() );
    ρ_t .Set( total_density, φc );
    Rv_s.Set( bulk_momentum_balance, Δτ(), ΔP_t(), ρ_t, g );

    // Bulk mass balance
    RP_t.Set( bulk_mass_balance, Δv_s(), P_t[0], P_f[0], φ[0] );

    // Fluid mass and momentum balance
    q_D .Set( darcy_flux, ΔP_f(), φc, g );
    RP_f.Set( fluid_mass_momentum_balance, Δq_D(), P_t[0], P_f[0], φ[0] );
};
```

Working on it!

GARNET


GPU
GPU
GPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
CPU
GPU
GPU
GPU

GARNET

- **We are using GPUs from Garnet!** Via C++ lambdas and Thrust

  Via class to manage both MPI3 shared memory and device memory (rank 0 dispatches GPU work)

  - Made GPU-aware version of `std::array (float3` [...] `)`
  - Based on `thrust::device_vectors` and pointers to host memory
- Status: Implementing whole User Interface functionality Thrust-aware
- After this week: merge CPU- and GPU-capable versions into one code.

GARNET