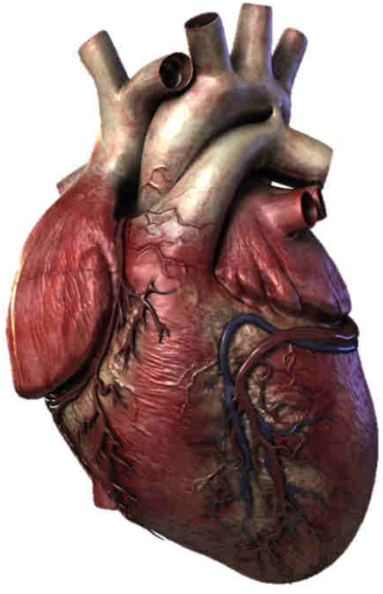
The human heart seen by the eyes of a computer scientist

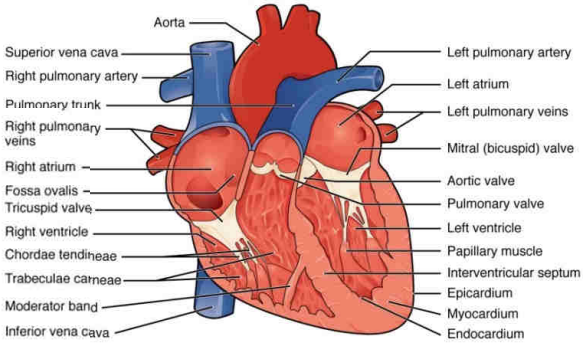
Marta Garcia-Gasulla, Alfonso Santiago, Guillaume Houzeaux, Beatriz Eguzkitza and Filippo Mantovani (BSC)

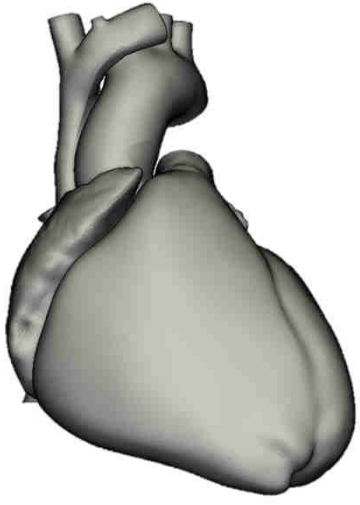
EU H2020 Centre of Excellence (CoE)

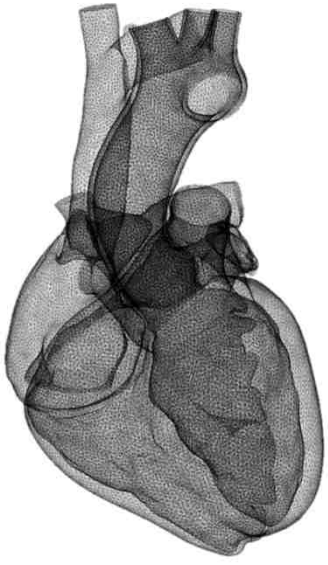
1 December 2018 – 30 November 2021

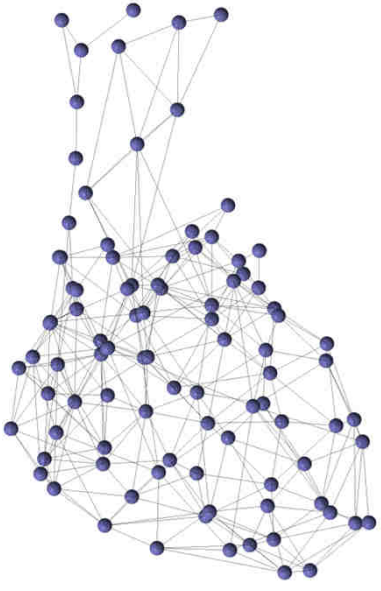
Grant Agreement No 824080

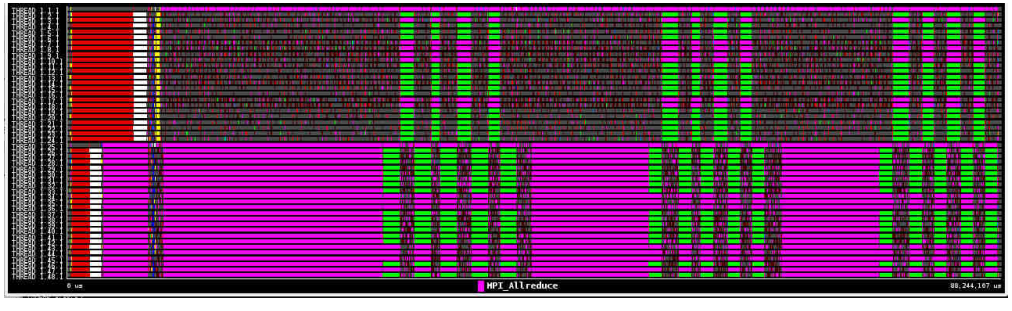



Aorta
Superior vena cava
Right pulmonary artery
Pulmonary trunk
Right pulmonary veins
Right atrium
Fossa ovalis
Tricuspid valve
Right ventricle
Chordae tendineae
Trabeculae carneae
Moderator band
Inferior vena cava
Left pulmonary artery
Left atrium
Left pulmonary veins
Mitral (bicuspid) valve
Aortic valve
Pulmonary valve
Left ventricle
Papillary muscle
Interventricular septum
Epicardium
Myocardium
Endocardium

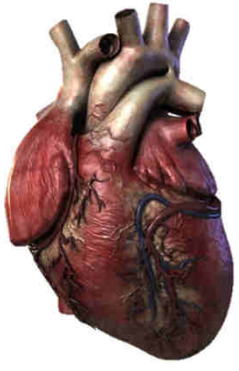







MPI_Allreduce

Who?



$$u^n = \frac{M}{\rho}(f_{vis} + f_{int} + B)$$

$$v^{n+\frac{1}{2}} = v^n + \frac{1}{2}\Delta t a^n$$

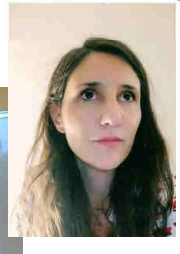
$$u^{n+1} = u^n + \Delta t v^{n+\frac{1}{2}}$$

$$\frac{\Delta\varphi^*}{\Delta t} + \mathbf{I}_{ion}(\varphi) = 0$$

$$\left(\frac{\mathbf{M}}{\Delta t} + \mathbf{G}\right)\Delta\tilde{\varphi} = -\mathbf{G}\varphi^*,$$

$$\varphi^{n+1} = \varphi^n + \Delta\varphi^* + \Delta\tilde{\varphi}.$$

$$\frac{\partial}{\partial x_j}\left(\left[1+\alpha^e\right]\frac{\partial b_i}{\partial x_j}\right) = C$$



Alya

HPCM

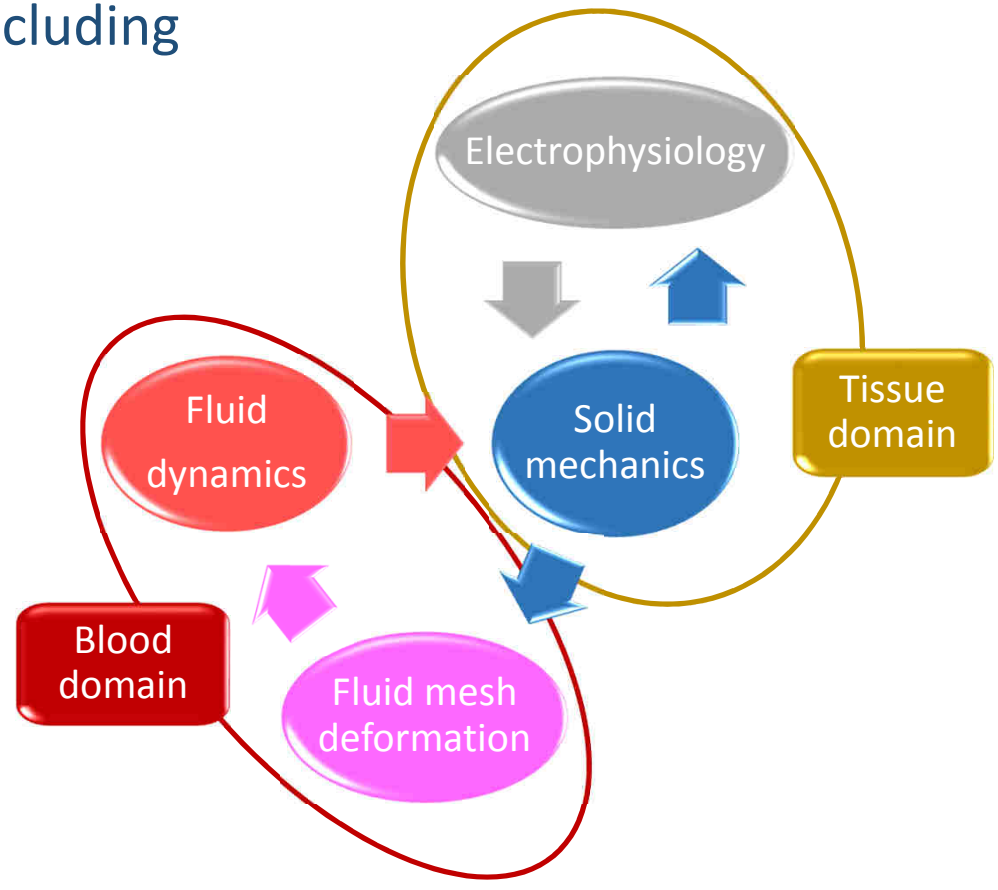
```
time: do while ( kfl_gotim == 1 )
   call Timste()
   reset: do
      call Begste()
      block: do while ( kfl_goblk == 1 )
         zone_coupling: do while ( kfl_gozon == 1 )
            call Begzon()
            coupling_modules: do while ( kfl_gocou == 1 )
               call Doiter()
               call Concou()
            end do coupling_modules
```

The simulation

- The heartbeat of a whole human heart including ventricles, atria and great vessels

- Tissue domain
  - **Electrophysiology**
    - Compute electrical impulse
  - **Solid mechanics**
    - Compute displacement of solid.

- Blood domain
  - **Fluid mesh deformation**
    - Compute mesh displacements
  - **Fluid dynamics**
    - Compute velocity and pressure of fluid

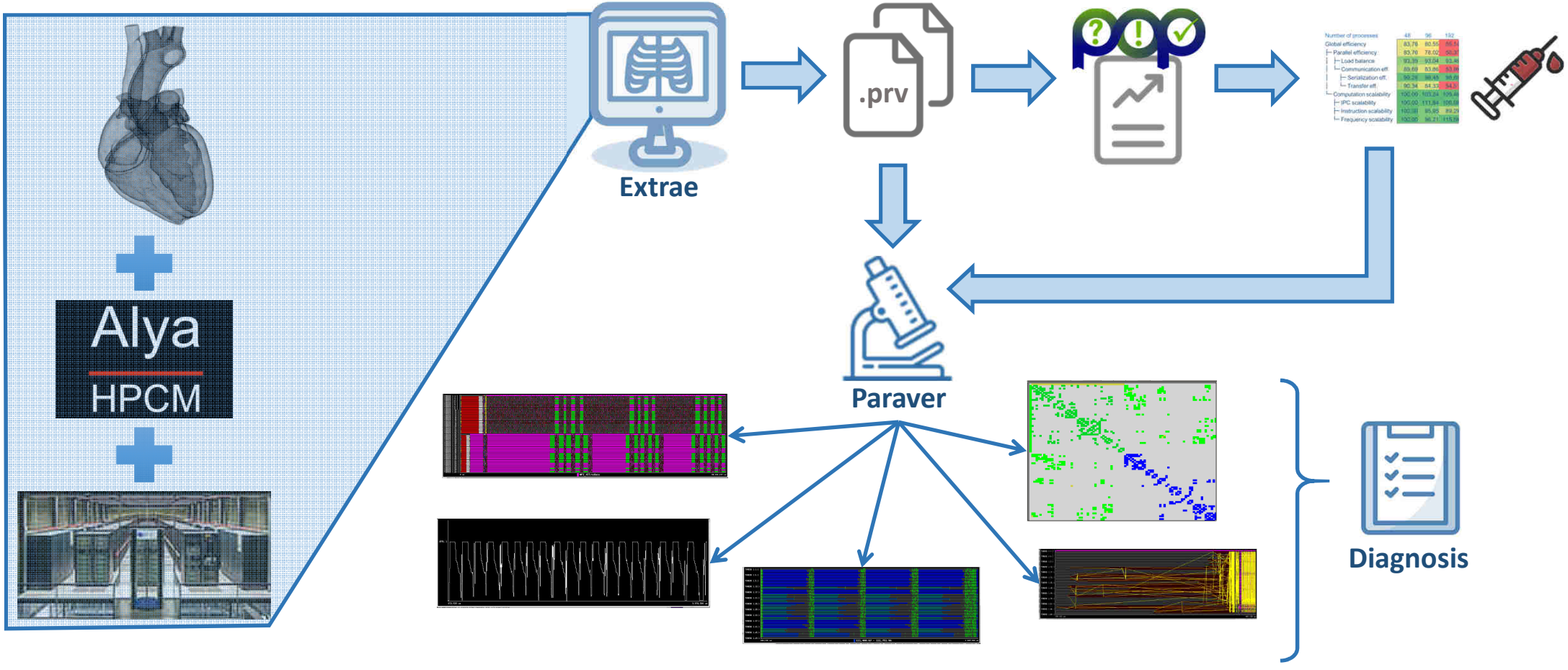


The simulation

- The code:
  - **Alya**: Is a simulation code for high performance computational mechanics
    - Unstructured Meshes
    - Parallelized using MPI and OpenMP
    - Developed at BSC
- The geometry: Two meshes
  - Tissue of the heart, 500k elements
  - Blood within the cavities, 400k elements

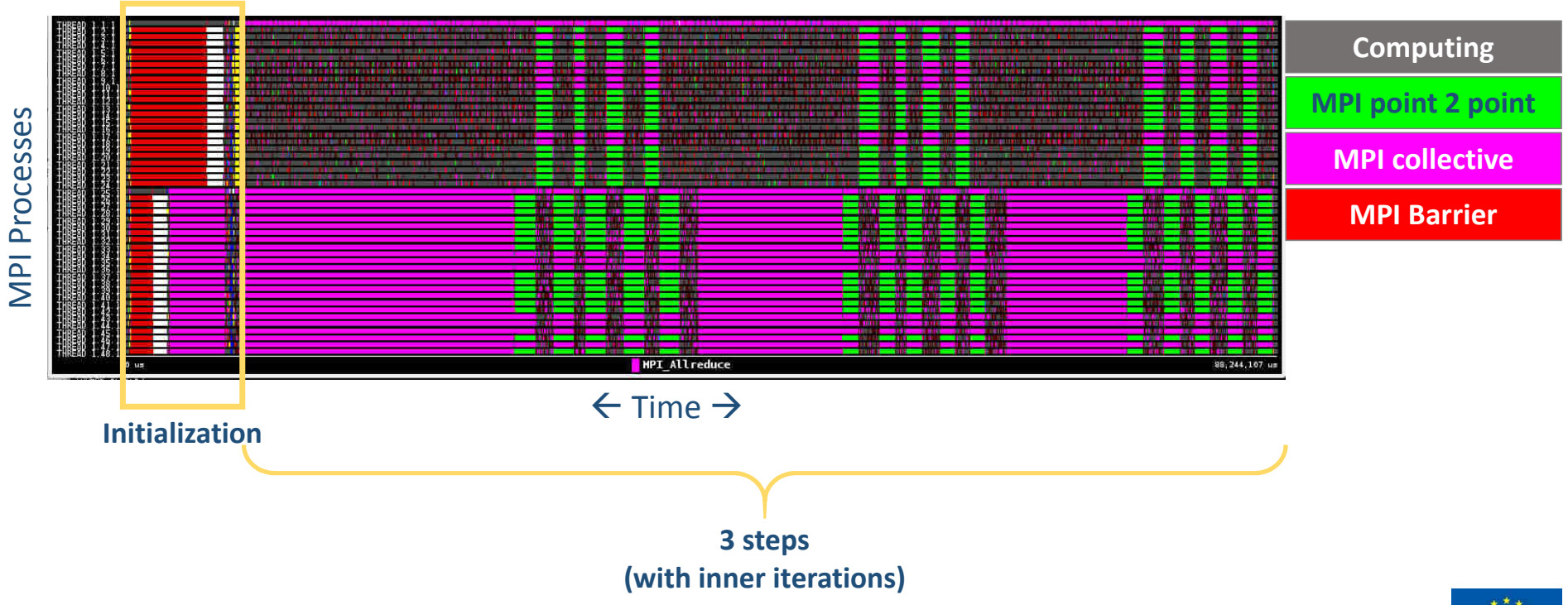
Alfonso Santiago et al. **Fully coupled fluid-electro-mechanical model of the human heart for supercomputers**. International journal for numerical methods in biomedical engineering (2018), 34(12), e3140.

How?

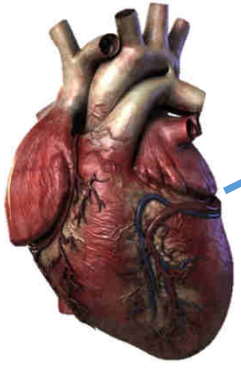


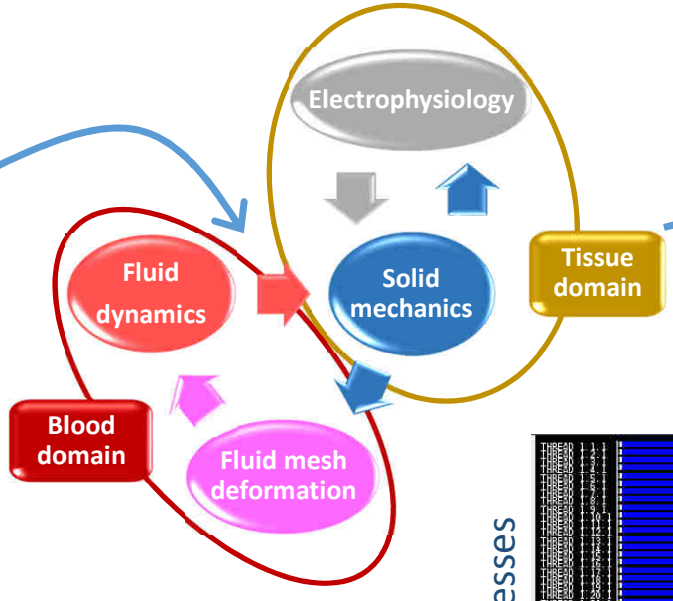
First view

- Check structure
  - ➢Find “interesting part” → Focus of Analysis (FoA)
  - ➢Usually iterative, without initialization and finish

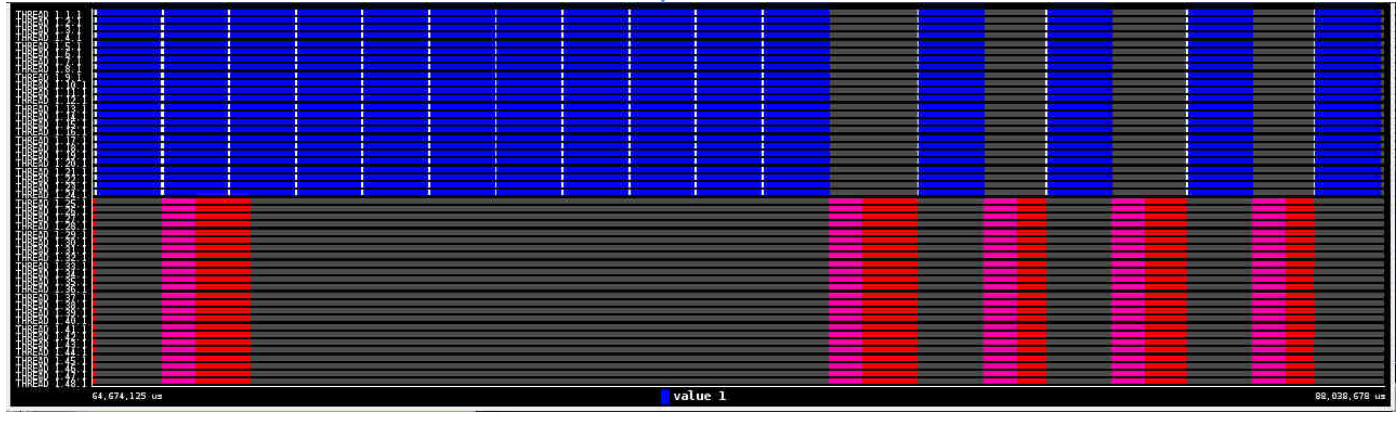


Where is the heart?





Electrophysiology
Solid mechanics
Mesh deformation
Fluid dynamics
Waiting

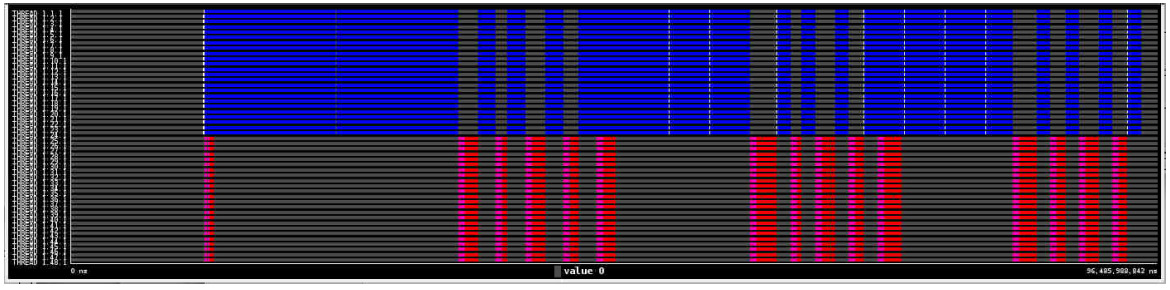


← Time →

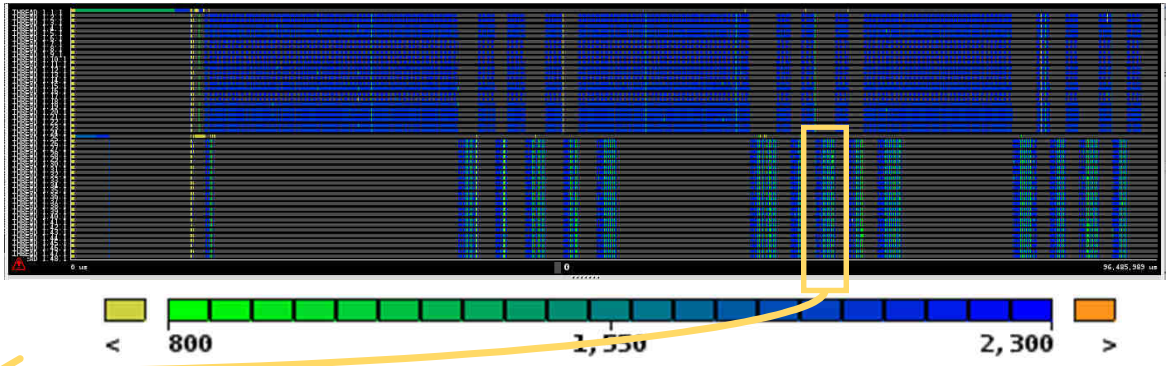
Measuring the heart rate

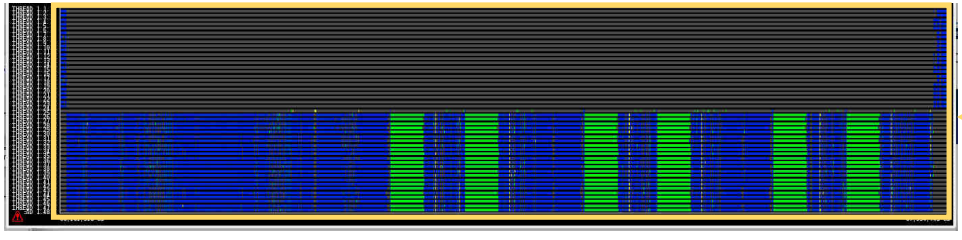
- One of the first sanity checks
  - Cycles per us

Phase



Cycles per us

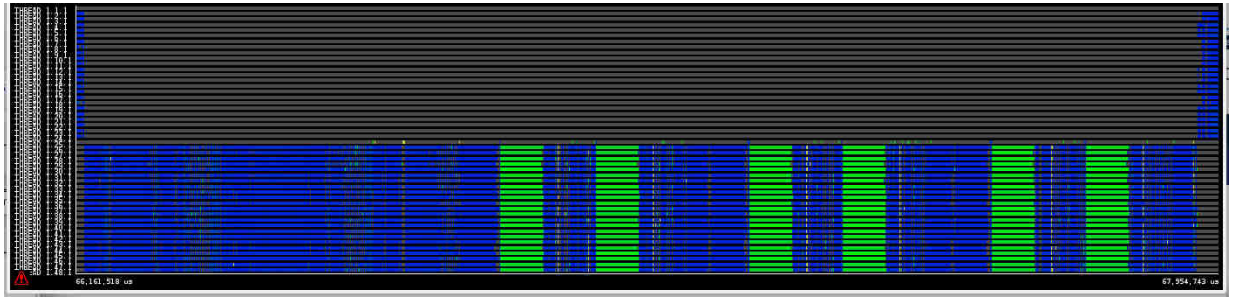




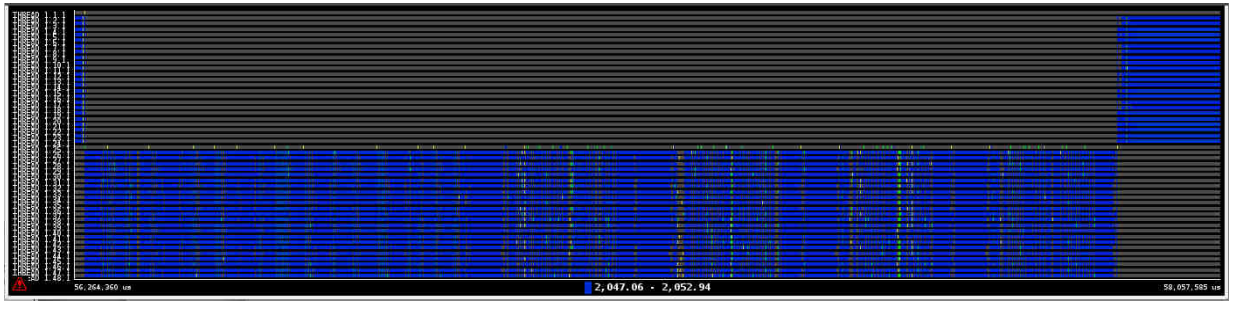
The patient has arrhythmia

- Why low value of cycles per us?
  - ➢ Digging in the code we find a large memory allocation and initialization that was not necessary

Cycles per us
With allocation
and initialization



Cycles per us
With allocation
and **NO** initialization



Efficiency metrics of the heart

Number of processes	48	96	192
Global efficiency	41,78	47,25	43,64
├ Parallel efficiency	41,78	40,02	33,27
│ ├ Load balance	57,53	59,95	60,32
│ └ Communication eff.	72,63	66,75	55,16
│ ├ Serialization eff.	73,83	69,81	65,38
│ └ Transfer eff.	98,37	95,62	84,37
└ Computation scalability	100,00	118,07	131,15
├ IPC scalability	100,00	111,64	116,10
├ Instruction scalability	100,00	107,29	109,11
└ Frequency scalability	100,00	98,57	103,53

Limiting factor for this number of cores

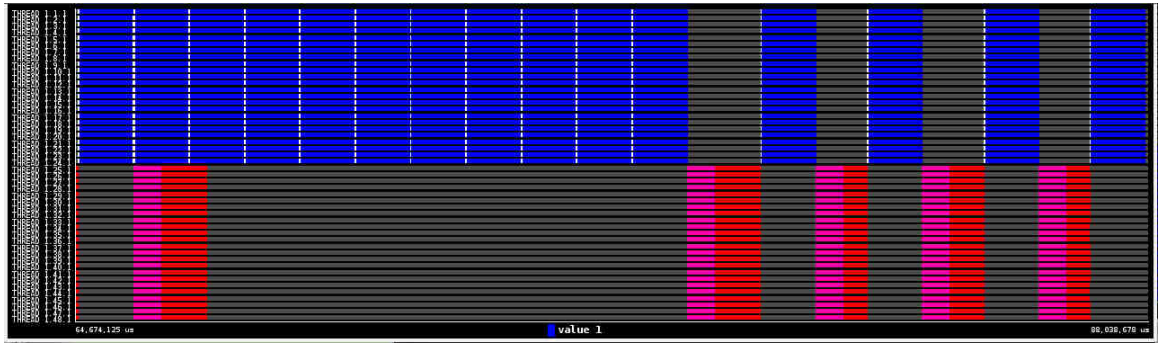
May become a limiting factor for higher number of cores

Good Scaling

- What I see:
  - Very low parallel efficiency
    - Only 33% of the time is used to do compute
  - High Load imbalance
    - 40% of time is “lost” waiting for more loaded processes
  - Serialization Problem

The diagnostic…

- When one physic is being solved the other must wait
- Common issue in Fluid-Structure-Interaction (FSI) problems
- How can we improve this?
  - Run both codes on the same cores
  - Assume MPI is not consuming cpu time
  - Use `DLB_node_barrier()`



  - More details…

Juan Carlos Cajas et al. **Fluid-Structure Interaction Based on HPC Multicode Coupling**
SIAM Journal on Scientific Computing 40 (6), C677-C703

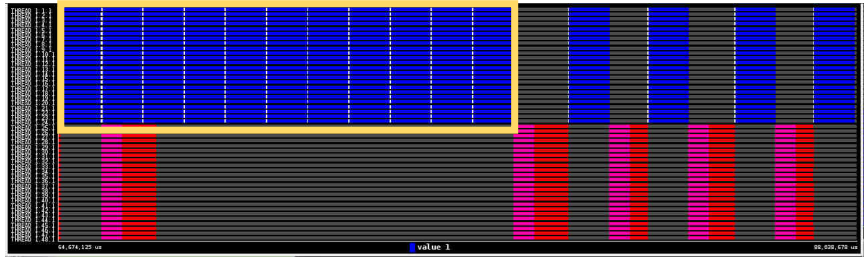
Efficiency metrics: Tissue

Number of processes	48	96	192
Global efficiency	75,50	89,77	96,44
├ Parallel efficiency	75,50	71,74	67,31
│ ├ Load balance	77,40	76,15	74,29
│ └ Communication eff.	97,54	94,21	90,60
│ ├ Serialization eff.	99,69	99,77	99,87
│ └ Transfer eff.	97,84	94,42	90,72
└ Computation scalability	100,00	125,14	143,28
├ IPC scalability	100,00	110,69	120,12
├ Instruction scalability	100,00	112,62	119,36
└ Frequency scalability	100,00	100,38	99,94

Limiting factor for this number of cores

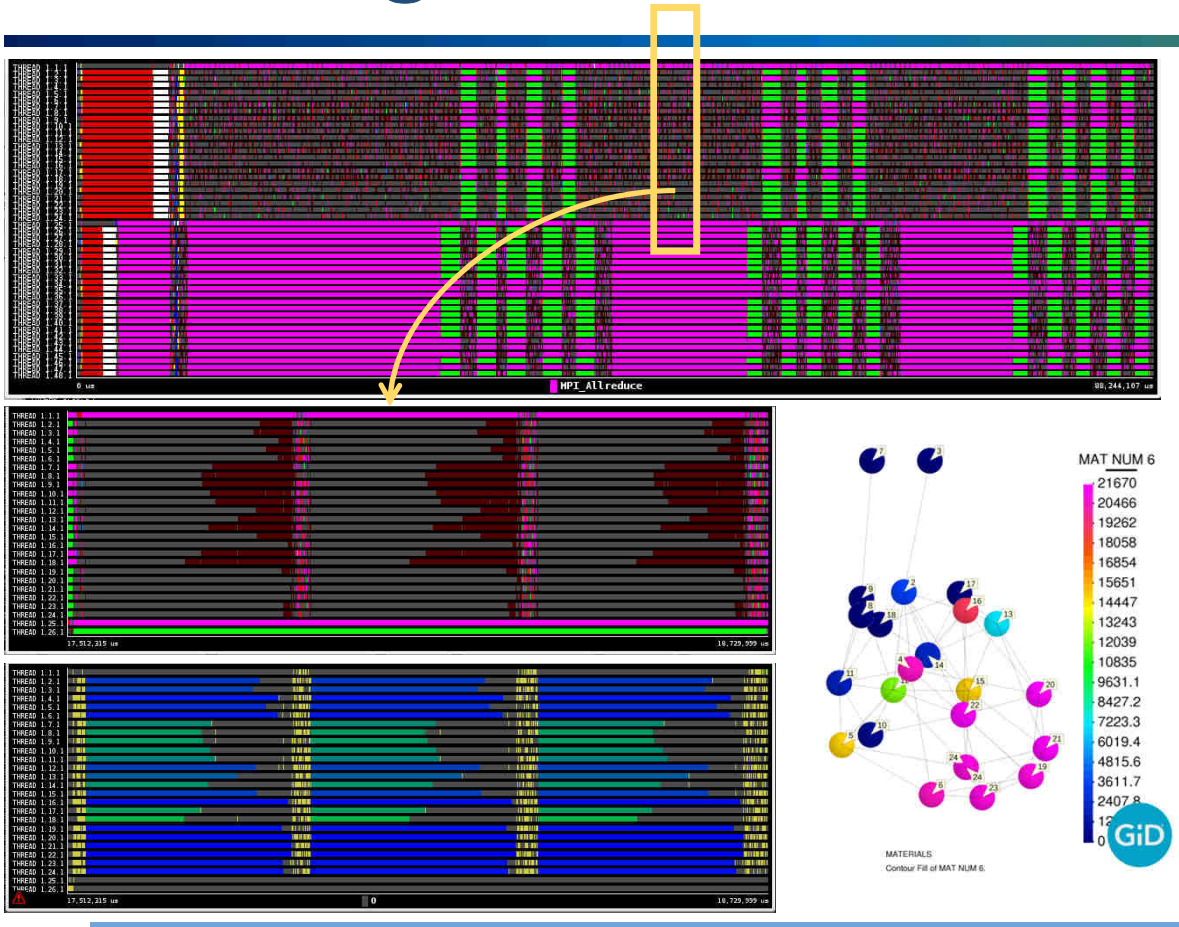
May become a limiting factor for higher number of cores

Good Scaling



- What I see:
  - Load Balance problem

The diagnostic for the tissue



- Where the load balance comes from?
  - ➢ Computational load imbalance
    - – Really some processes compute more than others
- Why?
  - ➢ Because of the different “materials”, some more “expensive”
- How can we improve this?
  - ➢ Partition better the mesh
  - ➢ Use Dynamic Load Balancing
  - ➢ More details…

Marta Garcia-Gasulla et al. **Runtime mechanisms to survive new HPC architectures: A use case in human respiratory simulations.** The International Journal of High Performance Computing Applications (2019):

Efficiency metrics: Blood

Number of processes	48	96	192
Global efficiency	83,76	80,55	55,14
├─ Parallel efficiency	83,76	78,02	50,37
│ ├─ Load balance	93,39	93,04	93,46
│ └─ Communication eff.	89,69	83,86	53,90
│ ├─ Serialization eff.	99,28	99,45	98,88
│ └─ Transfer eff.	90,34	84,33	54,51
└─ Computation scalability	100,00	103,24	109,46
├─ IPC scalability	100,00	111,84	106,08
├─ Instruction scalability	100,00	95,95	89,29
└─ Frequency scalability	100,00	96,21	115,58

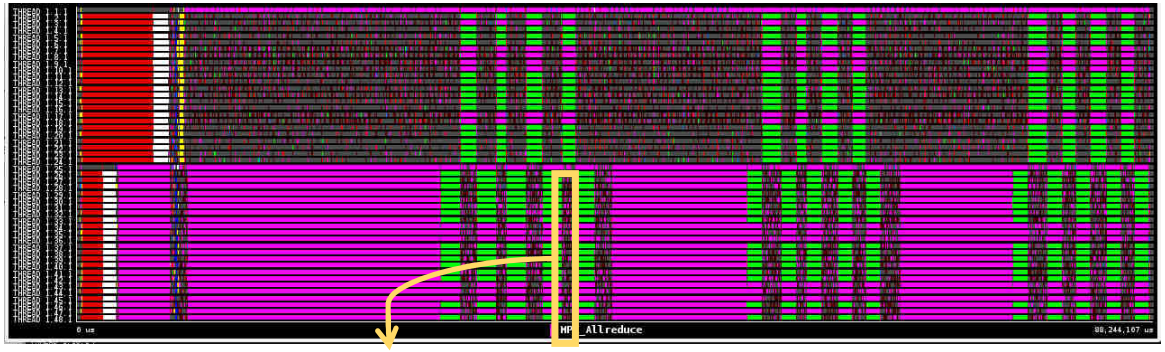
Limiting factor for this number of cores

May become a limiting factor for higher number of cores

Good Scaling

- What I see:
  - Very low transfer efficiency
    - Almost 50% of time in communication

The diagnosis for the blood



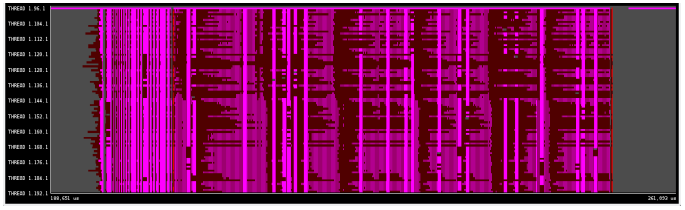
MPI calls in 800ms

	MPI_Barrier	MPI_Allreduce	MPI_Comm_rank	MPI_Comm_size	MPI_Sendrecv
Total	2,400	675,648	864	864	1,974,452
Average	25	7,038	9	9	20,783.71
Maximum	25	7,038	9	9	42,657
Minimum	25	7,038	9	9	6,094
StDev	0	0	0	0	7,384.27
Avg/Max	1	1	1	1	0.49

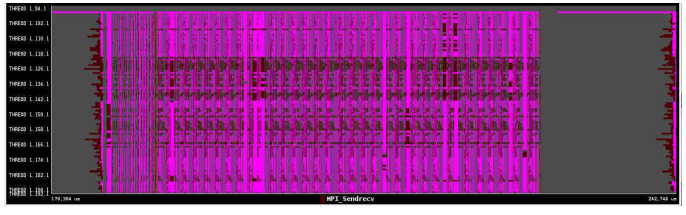
- High number of collectives
  - Synchronization problem
  - Depending on size → BW
- Very high number of point to point communications
  - 648 calls in 76 us
  - High imbalance in number of calls
    - Produces imbalance at collectives
- How can we improve this?
  - Can we reduce the number of point 2 point communications?
  - Can we partition with less neighbors?

Who to blame? Bandwith or Latency?

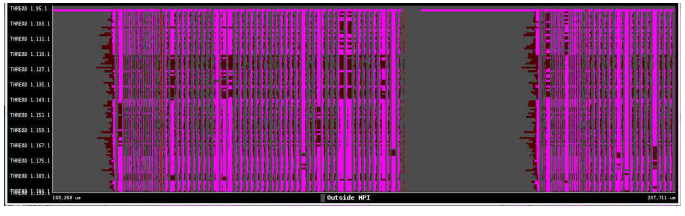
- Original



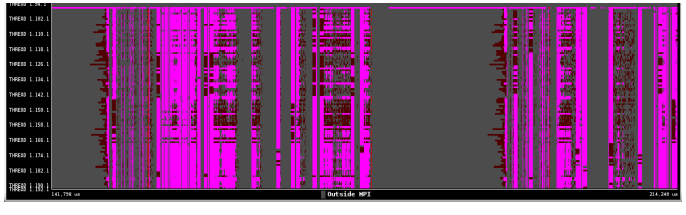
- Bandwidth = ∞



- Latency = 0



- Ideal



Lessons learned

- Interdisciplinary work
  - Enriches both sides
  - Advances both sides
  - Is FUN!!
- Performance analysis
  - MUST be done
  - Is useful to…
    - Improve performance
    - Find bugs
    - Find configuration problems…
  - Is FUN!!

Do you want more?

is a Center of Excellence
for Performance Optimisation and Productivity

- We do a performance analysis of your code/simulation/experiment… **Free of charge**
- Who can ask for it?
  - Any developer or user of a code from EU
  - From academic, research or commercial organisations
- How?
  - Fill the form in:
    `https://pop-coe.eu/request-service-form`
- Or e-mail me: `marta.garcia@bsc.es`

Performance Optimisation and Productivity

A Centre of Excellence in HPC

Contact:

https://www.pop-coe.eu

mailto:pop@bsc.es

@POP_HPC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 676553 and 824080.