CRESTA

Enabling In-situ Pre- and Post-Processing for Exascale Hemodynamic Simulations

− A Co-Design Study with the Sparse Geometry Lattice Boltzmann Code HemeLB

Fang Chen*, Markus Flatken*, Achim Basermann*, Andreas Gerndt*, James Hetherington+, Timm Krüger+, Gregor Matura* and Rupert Nash+

*German Aerospace Center (DLR)
+University College London (UCL)

Co-design with HemeLB: Overview (1)

- Objectives: development of in-situ pre- and post-processing as well computational steering tools to support HemeLB users:
  - Mesh manipulation and partitioning tools;
  - Visualisation tools;
  - Data management tools.

Co-design with HemeLB: Overview (2)

- **Task 1, pre-processing: partitioning and mesh manipulation**
  - DLR Cologne + UCL

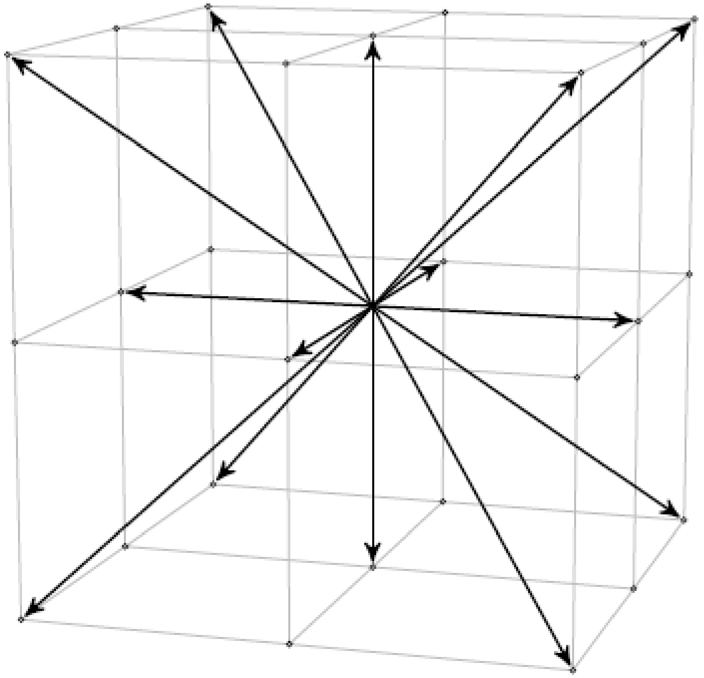
- **Task 2, post-processing: interactive data exploration and visualisation**
  - DLR Brunswick + UCL

- **Task 3, remote hybrid rendering for immersive visualisation and collaborative analysis**
  - University of Stuttgart

CREST

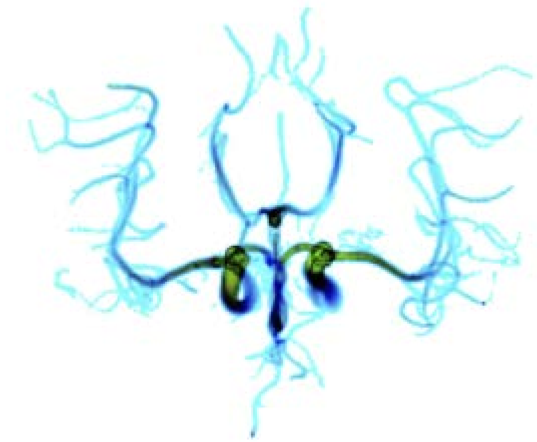
Introduction to Lattice Boltzmann

- A CFD solver
- Cubic lattice, no decomposition issue
- Good for parallelisation
- Good for complex geometry
- Easy to add new physics
- Collisions are local, stream is one-directional communication to neighbours



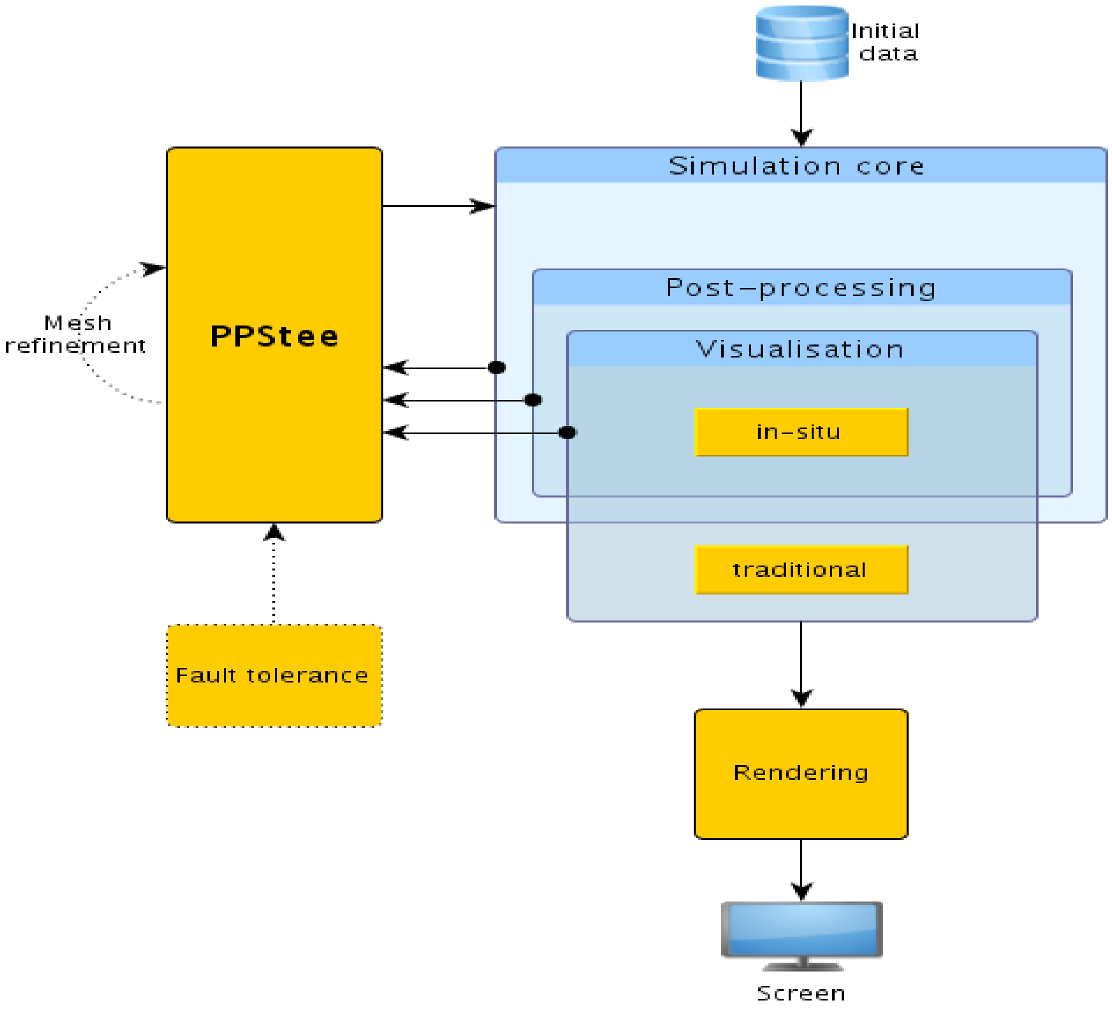
Introduction to HemeLB

- Aneurysm simulation
  - Fluid dynamics plays a role in the process of many diseases
  - The heart, the brain, tumours, drug delivery
  - Hemodynamics – blood flow – most notably



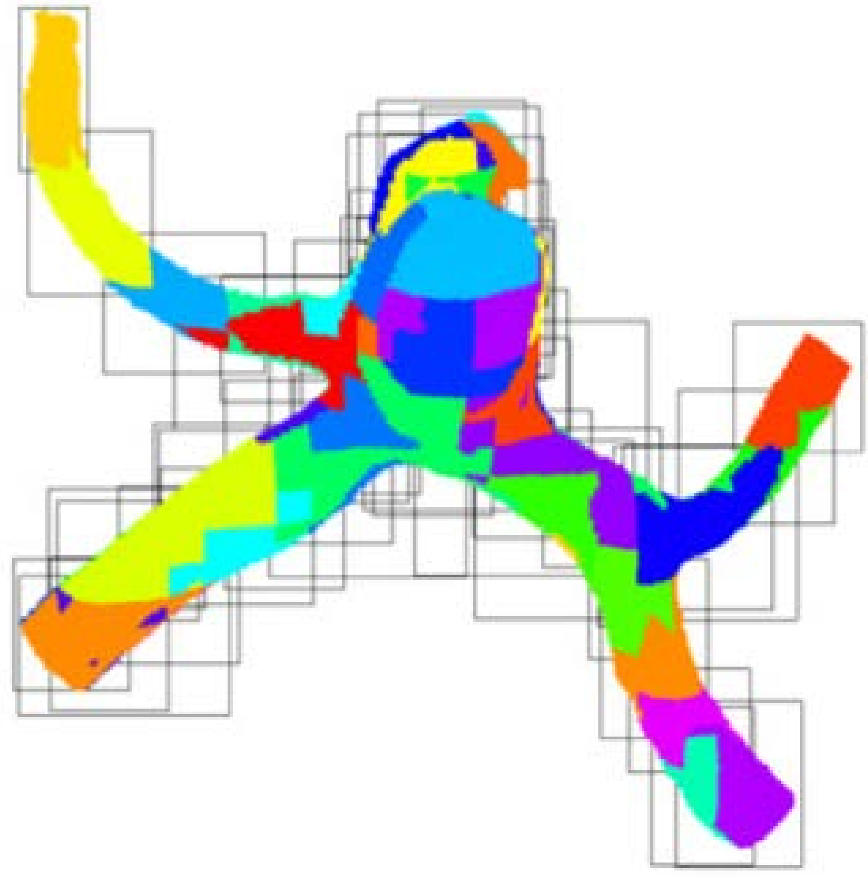
CREST

Co-design architecture



Pre-Processing

- HemeLB, background for pre-processing:
  - Hemodynamics solver for sparse geometries (typically 90% sparseness)
  - Aim: optimal load balancing
  - Requires advanced domain decomposition
  - Relies on a 3rd party partitioning library
- Current HemeLB partitioner: ParMETIS
  - High quality partitioner for large graphs
  - Fully parallel



CREST

Pre-Processing: Partitioners

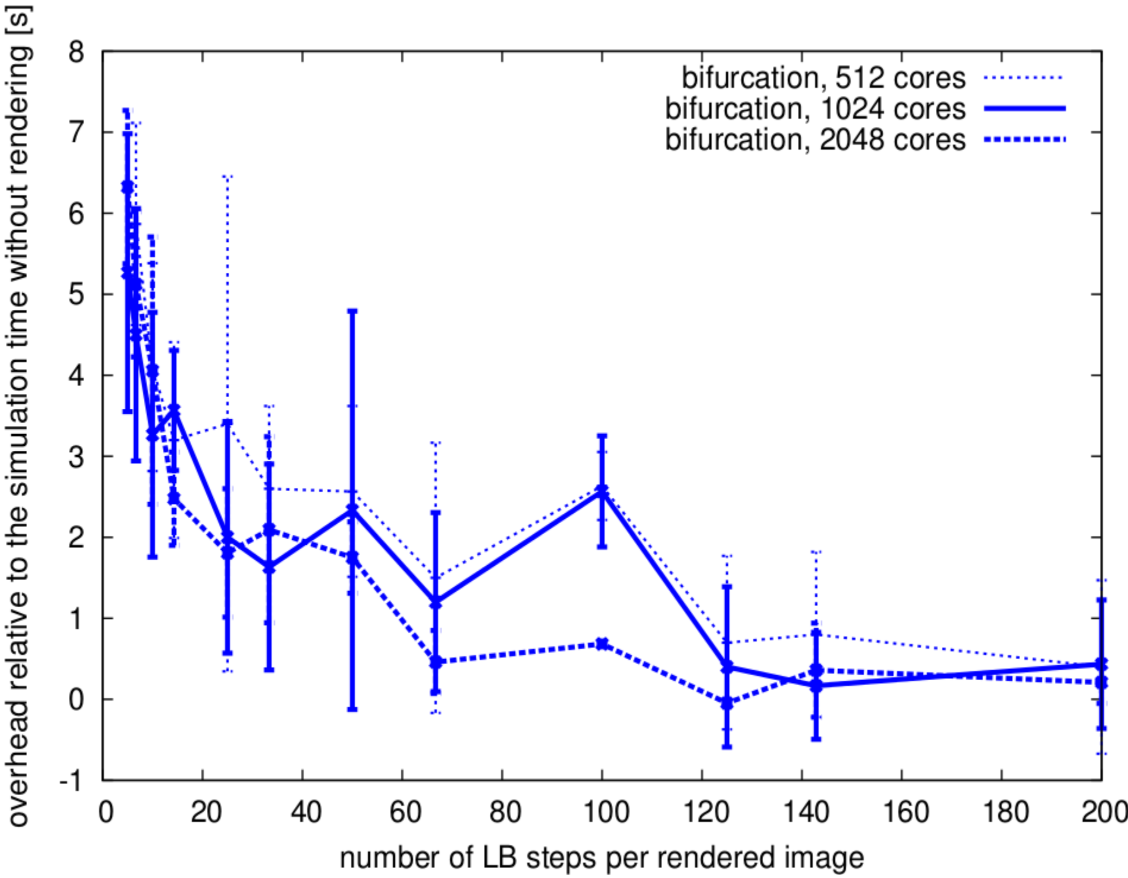
- Case-study: ParMETIS
- Data structures: graph data
  - Minimal memory requirements
  - Minimal inter-processor communication for global maintenance
  - Compatible to other data models by PTScotch, Zoltan, Boost
  - ➢ Suitable for exascale systems
- Algorithms
  - Multilevel k-way
  - Additional customisations possible:
    - adaptive mesh refinement
    - multi-phase partitioning
    - heterogeneous architecture matching
  - Especially multi-phase is useful for exascale systems to consider costs of different simulation parts appropriately in order to achieve load-balance

Pre-Processing: HemeLB Partitioning Discussion

- Case-study: HemeLB
  - Initial decomposition (simple)
  - Optimised reading (more than one yet not all cores) and distributing
  - Optimised decomposition by ParMETIS
  - Re-distribution
- Regarding exascale
  - Initial decomposition: no data duplication
  - Read-in: parallel yet adaptive to filesystem, thus no I/O bottleneck and small fragmentation
  - Partitioning: does not yet account for post-processing

Post-Processing

- HemeLB post-processing, status:
  - Large-scale simulations involve huge datasets
  - Storing whole datasets is impractical
  - Data extraction on subdomains possible (plane, line, surface, point), but has to be specified before the simulation
  - Strategy: in-situ rendering
  - HemeLB shows excellent performance for overlapping computation and rendering



Simulation time: 31.4, 16.1 and 7.81 s on 512, 1024 and 2048 cores, respectively

CREST

Post-Processing: Techniques and Data Structures

Issues

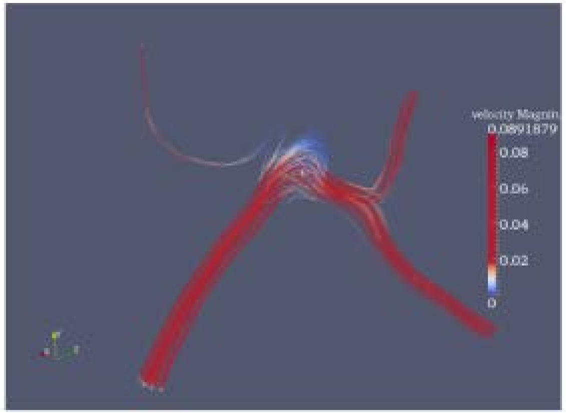
- Data reduction
- Memory layout
- Parallelisation with respect to visualisation algorithms
- How to become interactive
- How to enable in-situ

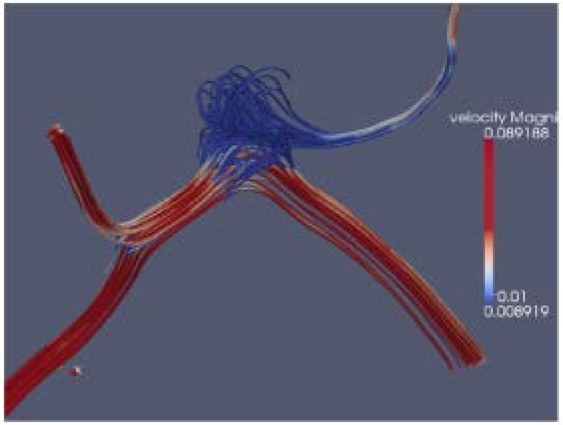
Roadmaps

- Hierarchical data structure
- Load/store on demand
- Query driven visualisation
- Visualisation techniques suitable for parallelisation

Post-Processing: Techniques and Data Structures

- In-situ processing
  - Interactive visualisation
  - On-the-fly visualisation, avoid writing out data
- Mesh-handling
- Data hierarchy
  - Fast data access
  - Multi-resolution data structure
- Visualisation techniques: volume rendering, line integrals, particle tracing …

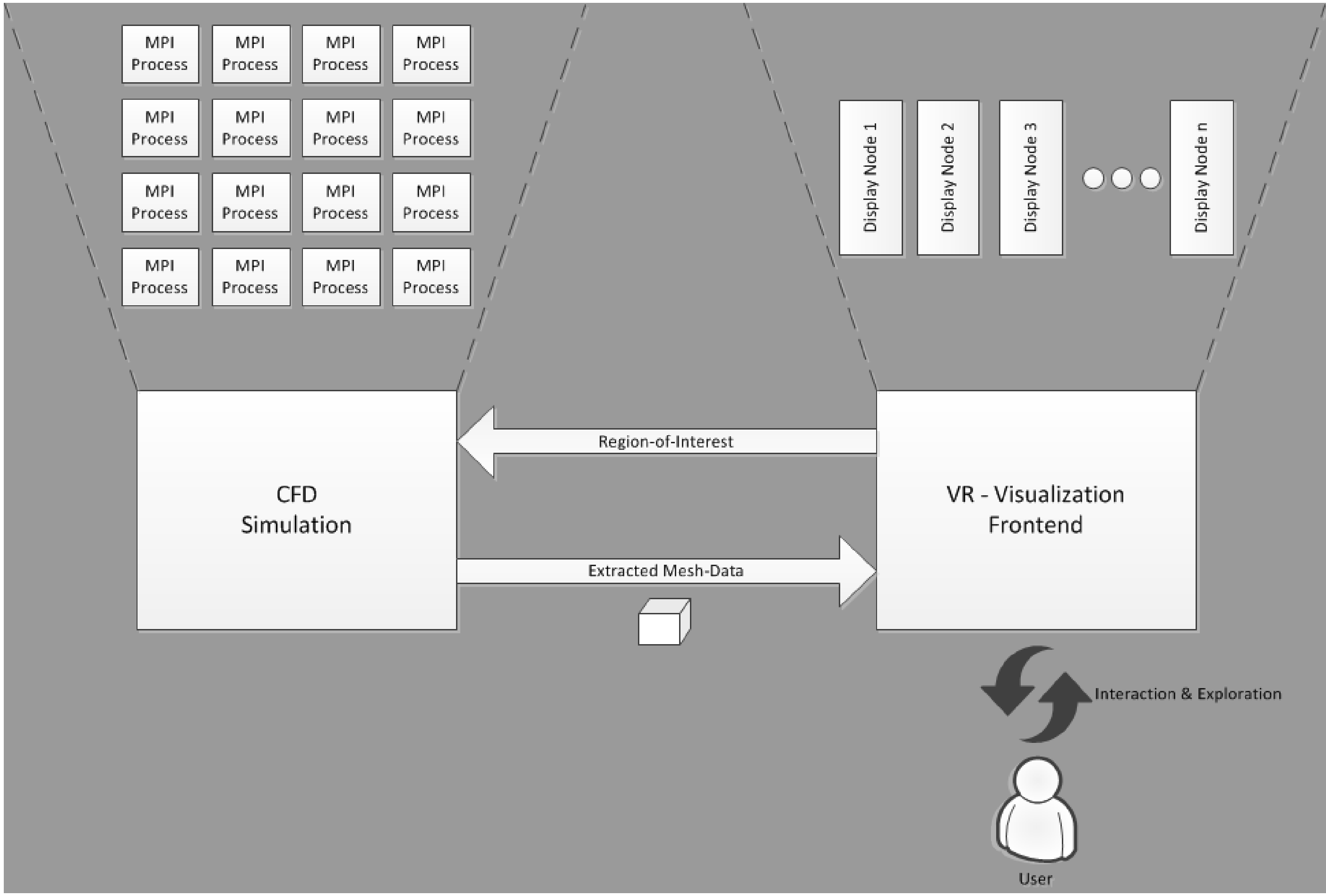




CREST

Post-Processing: System Survey

- Query driven, visualisation techniques
- Data hierarchy
- Level of details
- Down-sample data
- Region of interest



CREST

Conclusions

- Pre-processing data structures and algorithms suitable for exascale systems identified.
- Prototype pre-processing interface for exascale systems defined.
- Concepts for data structures and algorithms suitable for exascale interactive in-situ post-processing developed.
- Tests of prototype post-processing tools with HemeLB data sets successful.

Future work

- Pre-processing: development of a pre-processing interface to expand load-balancing beyond core computation, i.e., to be sensitive to post-processing, data initialisation and a fault tolerance framework
- Post-processing: development of a scalable visualisation system. Focus will be on identifying and developing visualisation methods that scale to large data sets.
  - The visualisation techniques used should provide meaningful representation of the simulation output, which can help specialists to analyse their data.
  - The visualisation system should provide feedback to the simulation which indicates where and how the simulation can be modified or refined.
- Enable computational steering: using visualisation output as a feedback, to modify and steer running simulations, in order to achieve a better simulation output.