Petascale Adaptive Computational Fluid Dyanamics

K.E. Jansen, M. Rasquin

Aerospace Engineering Sciences
University of Colorado at Boulder

O. Sahni, A. Ovcharenko, M.S. Shephard, M. Zhou, J. Fu, N. Liu, C. Carothers

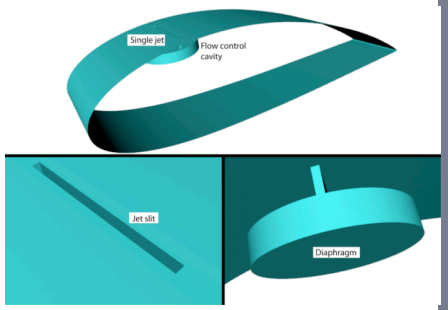
Scientific Computation Research Center

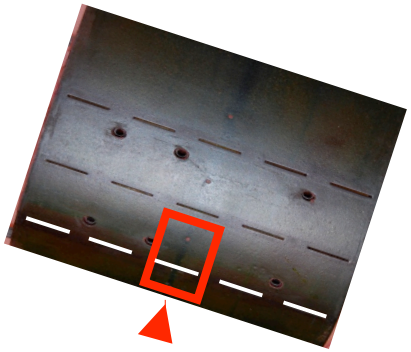
Rensselaer Polytechnic Institute
Department of Mechanical, Aerospace and Nuclear Engineering

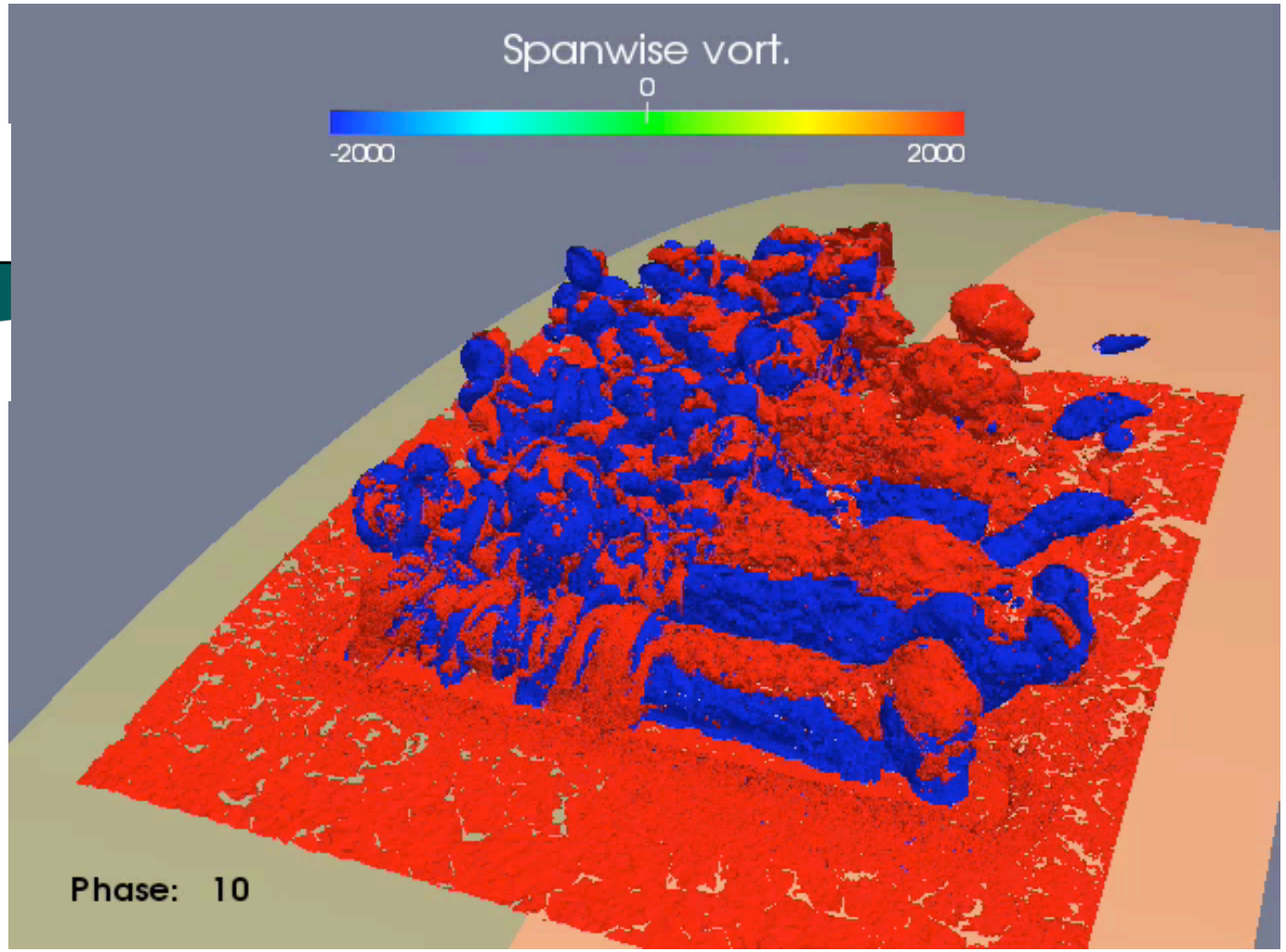
Sponsors: DOE: SciDAC-ITAPS, NERI; NSF: PetaApps, ITR, CTS; AFOSR, IBM, Northrup Grumman, Boeing, Lockheed Martin, Motorola
Computer Resources: INCITE (ANL, ORNL), TeraGrid (TACC,NICS), JSC, RPI-CCNI

Problems of Interest

Aerodynamic Flow Control

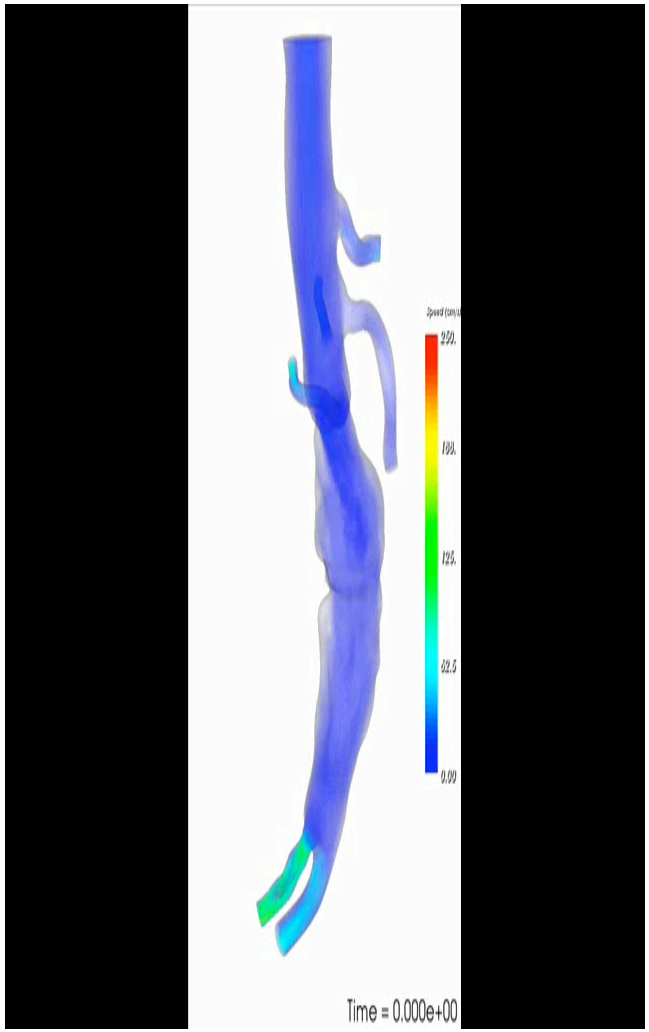






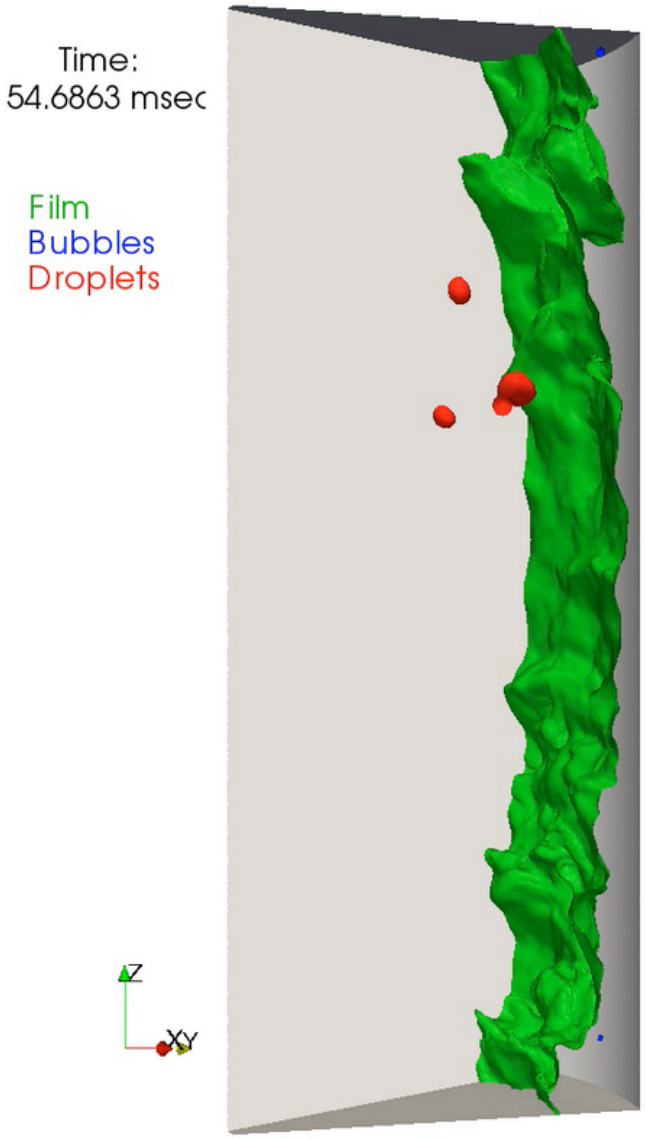
Problems of Interest

Cardiovascular flow: Abdominal Aorta Aneurysm



Problems of Interest

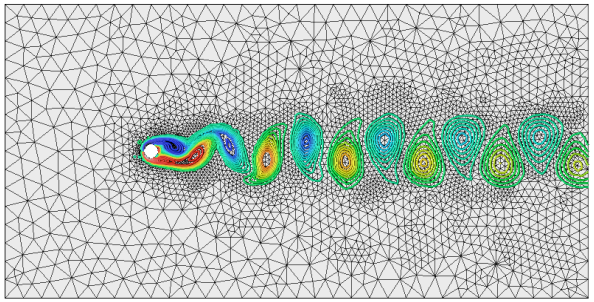
Two-phase
Annular Flow
(level set method)



PHASTA Flow Solver

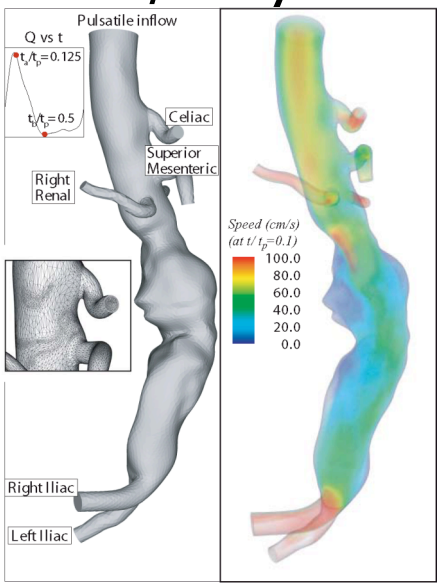
- Parallel Hierarchic Adaptive Stabilized Transient Analysis

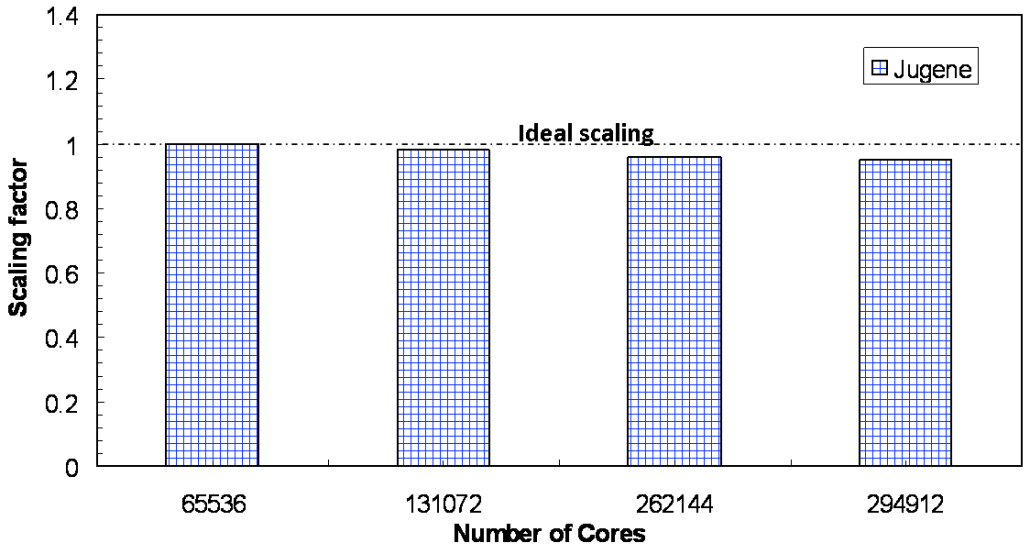
- Stability with Accuracy
  - Time accurate stabilized finite element method
  - Hierarchic spatial basis (currently p<4) $O(h^{p+1})$
  - Time integration: implicit (2^{nd} order generalized alpha method).
- Adaptivity
  - Grid matches physical scale
  - Anisotropic and transient
- Parallel
  - Excellent scaling to 288k processors (95%)



Strong Scaling – 5B Mesh up to 288k Cores

- AAA 5B elements: full-system scale on Jugene (IBM BG/P system)





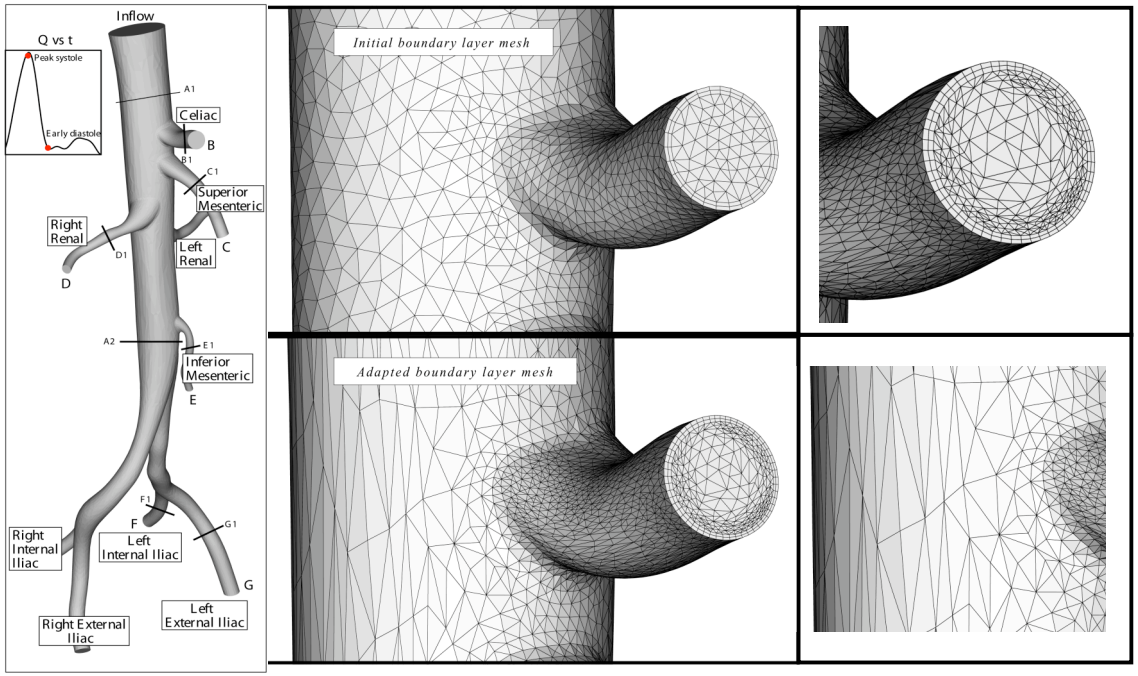
5B elements mesh (Jugene:IBM BG/P)		Eqn. form.		Eqn. soln.		Total	
num. of core	avg. elem./core	time	s-factor	time	s-factor	time	s-factor
65,536 (base)	76,480	119.64	1	162.59	1	288.23	1
131,072	38,240	59.69	1.00	84.09	0.97	143.78	0.98
262,144	19,120	30.02	1.00	43.24	0.94	73.26	0.96
294,912	16,995	26.71	1.00	39.15	0.92	65.86	0.95

without IPMod strong scaling factor is **0.88** (time is 70.5 secs),
for production runs savings can be in **43 cpu-years**

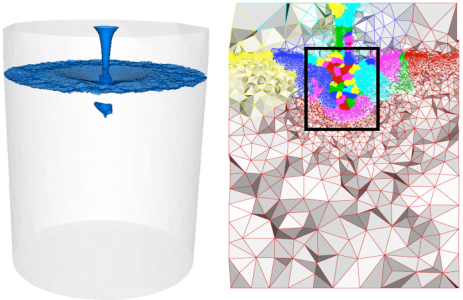
Current Approach – Unstructured Meshes

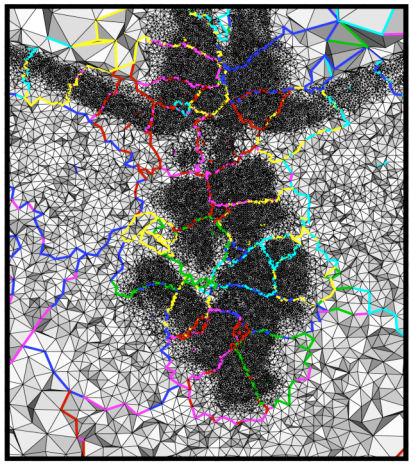
- Flow solver is designed to use general unstructured, anisotropic meshes; current meshes are constructed using parallel adaptive meshing capable to handle arbitrary complex geometry:

Healthy human aorta
(with adapted boundary layer mesh)

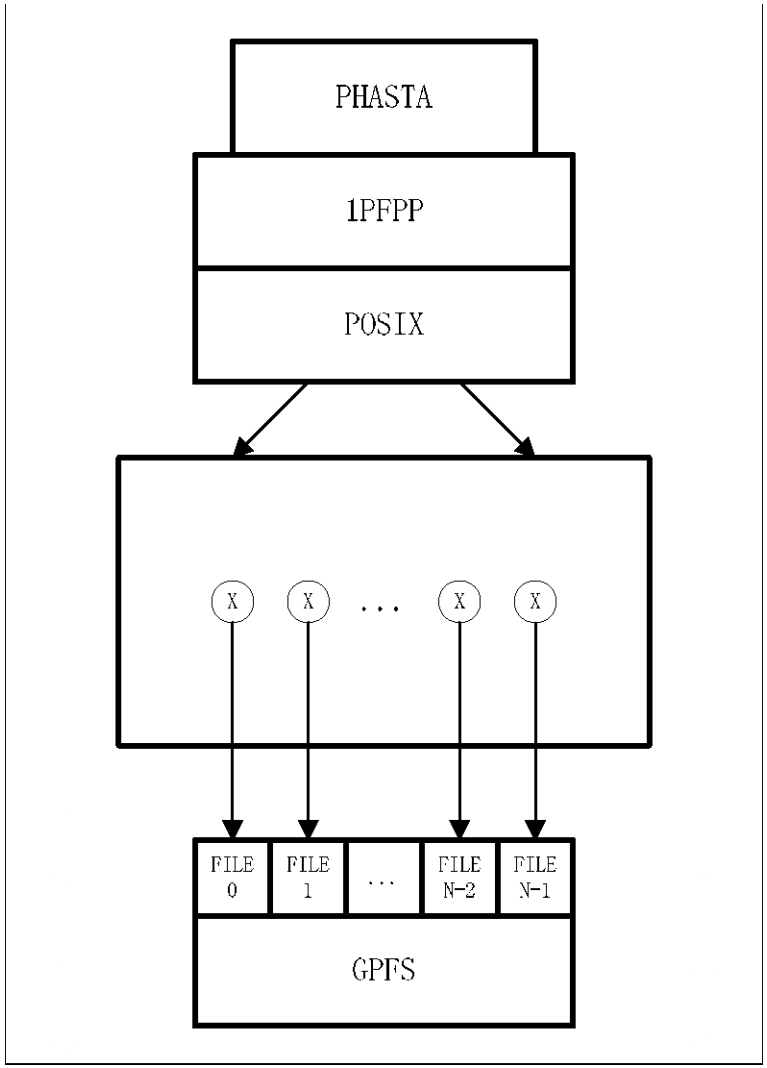


Plunging liquid jet
(colors indicate parts of a partitioned mesh)



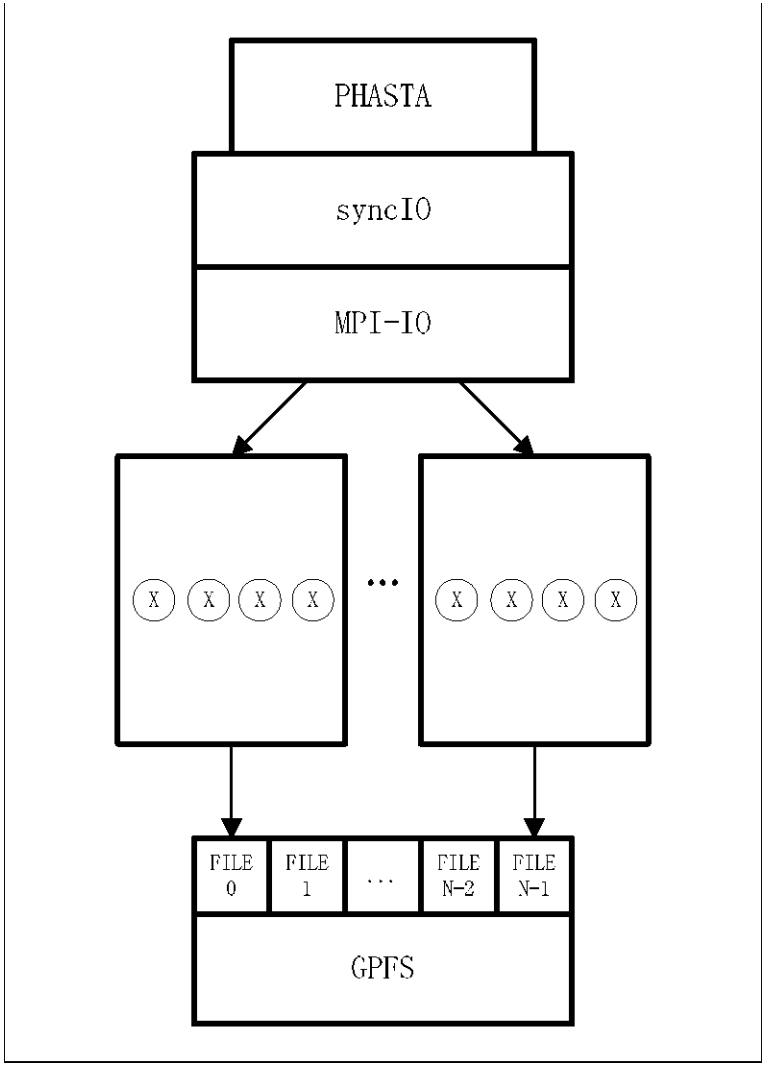


OLD I/O: 1 POSIX File Per Processor



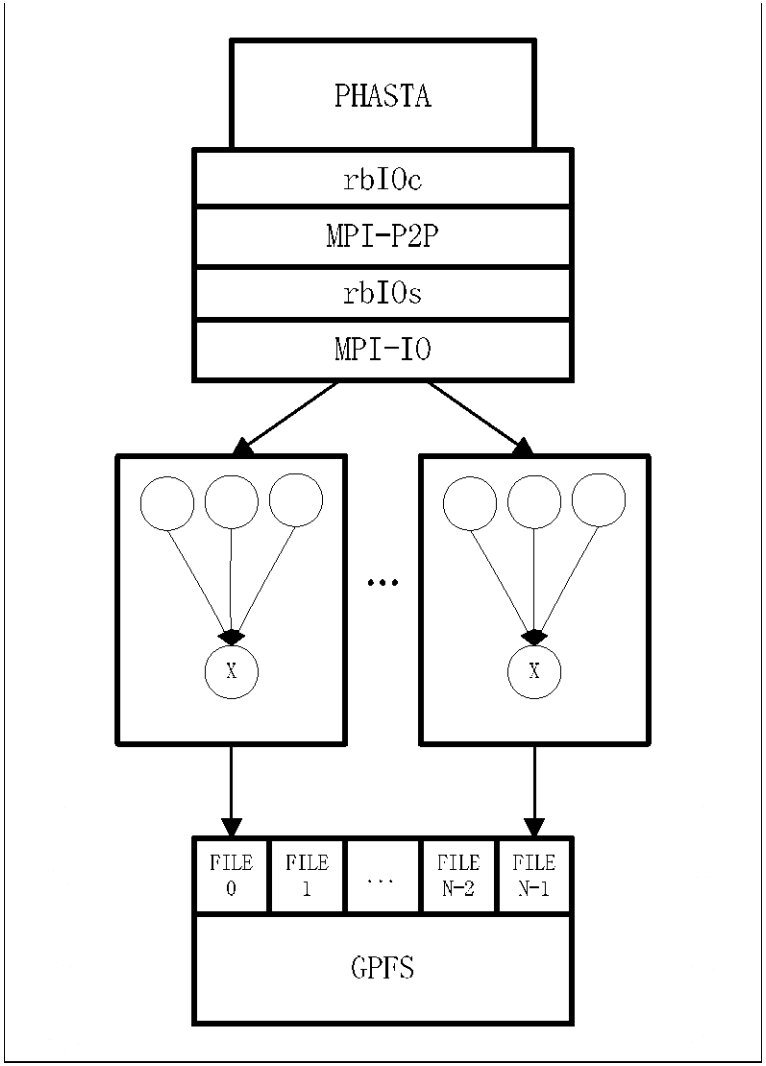
- Pros:
  - parallelism, high performance at small core counts
- Cons:
  - lots of small files to manage
  - LOTS OF METADATA – stress parallel filesystem
  - difficult to read back data from different number of processes
  - @ 300K cores yields 600K files
    - @ JSC ➔ kernel panic!!
  - PHASTA currently uses this approach…
- < 1 GB/sec on BG/L

I/O: MPI_File alternatives and syncIO



- Flexible design allows variable number files and procs/writers per file
- Within a file, can be configured to write on "block size boundaries" which are typically 1 to 4MB.
- Implemented using collective I/O routines : e.g., MPI_File_write/read_at_all_begin
- BG/P: 11.6 GB/sec read,
  25 GB/sec write

I/O: MPI_File alternatives and rbIO



- Rb ➔ "reduced blocking"
- Targets "checkpointing"
- Divides application into workers and writers with 1 writer MPI task per group of workers.
- Workers send I/O to writers over MPI_Isend and are free to continue –
  - e.g., hides the latency of blocking parallel I/O
- Writers then perform blocking MPI_File_write_at operation using MPI_COMM_SELF communicator
- BG/P: ~18 GB/sec actual write,

~167 TB/sec perceived write @ 128K cores

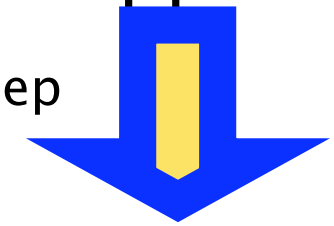
- So long as the workers do not request another write before writers complete, there is a big efficiency gain.

  Writer pool can be sized to achieve this.
- Can we message directly I/O nodes?

In situ vizualisation

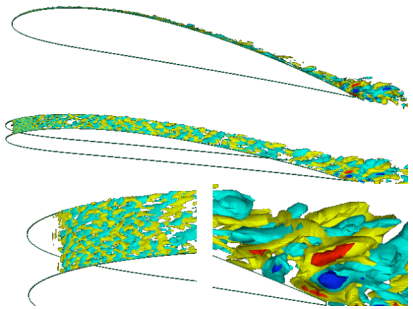
- Kernel of ParaView has been linked into PHASTA on N processors.
- Filter chain defined on interactive client and saved
- Filter chain executed at user prescribed frequency
    - Current solution, coordinates and connectivity passed from PHASTA to ParaView
    - ParaView executes filter chain on N processors, collects filter output geometry and transfers (by sockets or files) to PVServer running on n cores of a viz cluster
    - Geometry rendered on viz nodes and displayed on a local ParaView Client
- Currently scales reasonably well on N=8k cores on BGL at CCNI
- Work underway to extend to Intrepid-Eureka with N=160k
- Should work for VisIt too.

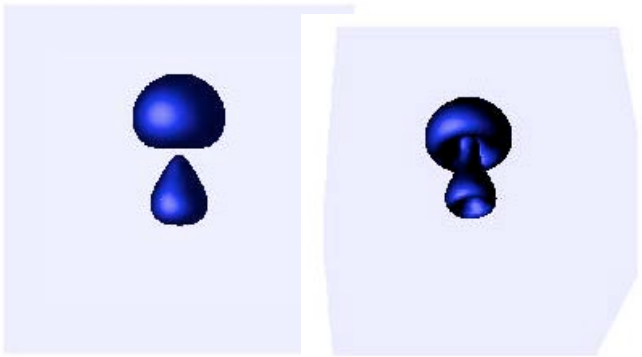
Summary

- Complex geometry/physics=> Real world Apps
- Implicit solvers: Complexity  but n_{step} 
- Excellent scaling results
- Big Science AND FAST SCIENCE
- Anisotropic Adaptivity brings real geometry problems into reach of solution in a USEFUL time frame
- Multiphase simulation capable of modeling turbulent flow with mixture of steam and water
- Complex geometry of very small flow control devices being simulated and validated.
- Patient-specific cardiovascular flows can be solved in clinically relevant time frame.

PHASTA Models

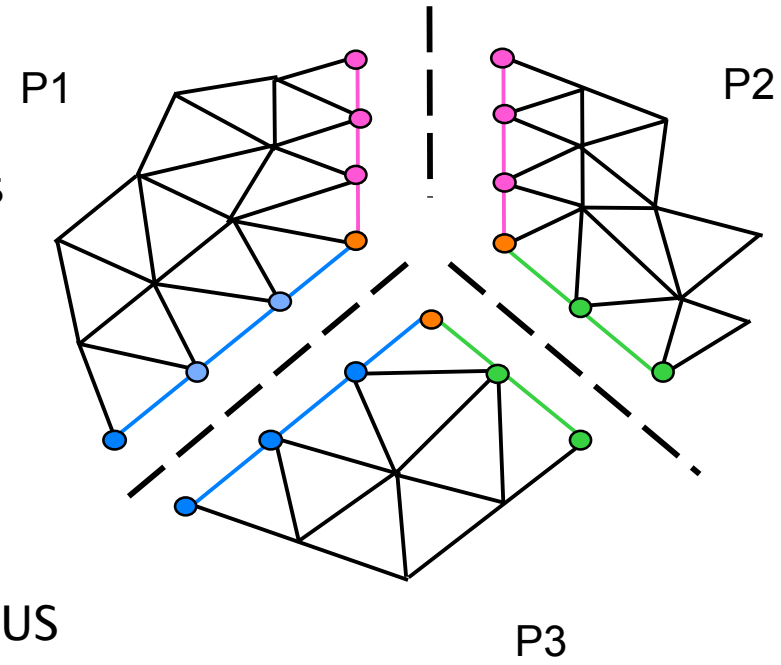
- Compressible or Incompressible flow solver
- Turbulence
  - Direct Numerical Simulation (DNS)
  - Large-Eddy Simulation (LES)
  - Reynolds-Averaged Navier-Stokes (RANSS)
  - Detached Eddy Simulation (DES) and other hybrid models





PHASTA Flow Solver Parallel Paradigm

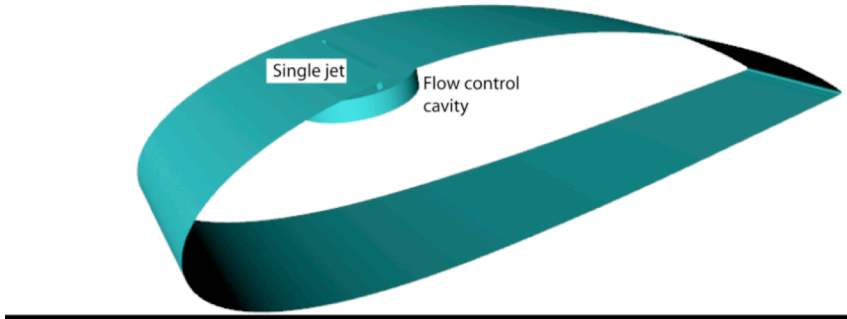
- Implemented in Fortran/C++ & MPI
- Input partitioned on a per-processor basis
- Unstructured mesh "parts" mapped to cores
- Two types of work:
  - **Equation formation (Ax=b)**
    - Matrix assembled on processor ONLY
    - O(40) peer-to-peer non-blocking comms to update share dofs in **b**
    - Scales well on many machines
    - Cost ÷ number of elements
  - **Implicit, iterative equation solution**
    - For each Krylov vector:
      - q=Ap (matrix-vector products)
      - Same peer-to-peer comm as **b** for **q** PLUS
      - Orthogonalize against prior vectors
      - **REQUIRES NORMS=>MPI_Allreduce**
      - Cost ÷ number of nodes

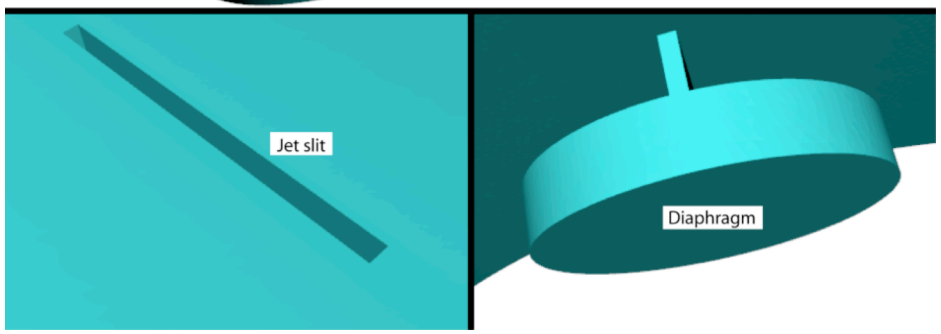


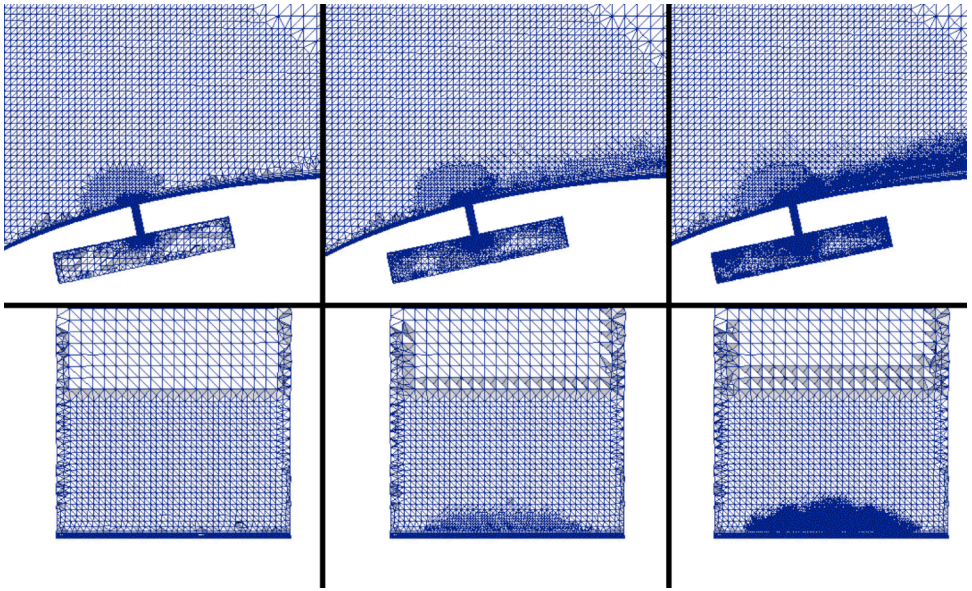
- **I/O**: Initial read of mesh data + Checkpoint after "N" interations

Finite-span synthetic jets

CAD geometry used in CFD matches experiment







Initial and 2 cycles of adapted mesh
Spanwise slice (top)
Streamwise slice 20 slit widths down stream (bottom)