OpenFOAM: Project Status and Plans for the Future

Hrvoje Jasak

h.jasak@wikki.co.uk, hrvoje.jasak@fsb.hr

Wikki Ltd, United Kingdom

Faculty of Mechanical Engineering and Naval Architecture

University of Zagreb, Croatia

Seoul National University, 14 April 2011

Outline

Topics

- Introduction and Project Overview
- Robustness and accuracy and scaling improvements
- New Developments
- Python and SALOME interface: VulaSHAKA project
- Deployment of OpenFOAM
- Recently deployed features and the upcoming release
- Community organisation
- Planned work and work-in-progress
- Summary and Outlook

Introduction

OpenFOAM

- A successful period for the community: further penetration into scientific research community and significant impact in industrial CFD
- Major community contributions: from new ideas to complete capability libraries
  - Radial Basis Function
  - Turbomachinery features and validation
  - Naval hydrodynamics effort and Overset grid
  - Python and SALOME integration
  - Robustness and accuracy improvements
- Next stage: **integration, consolidation, validation**
  - **OpenFOAM Foundation**: formalising the legal side of the project
  - Public test loop, nightly builds, validation cases
  - Documentation and community portal
  - Integrated cross-platform version
  - Improved release schedule

Robustness, Accuracy, Scaling

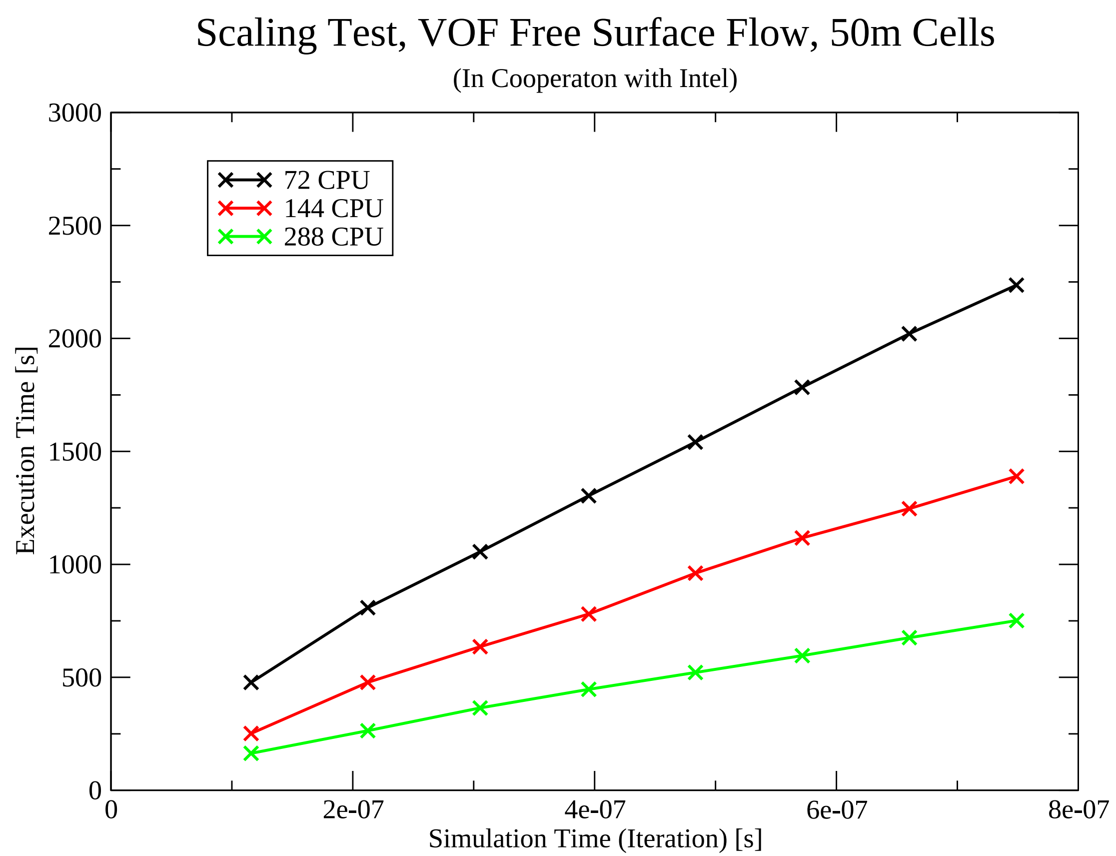
Robustness and Accuracy Improvements

- Tetrahedral mesh solution improvement
  - Historically, tet meshes are easier to generate, but “hex-based” CFD methodology has produced poorer results, especially for boundary layers. Improvements took approx. 6 years in Star-CD and Fluent
  - Pointwise mesh generator: unstructured complex geometry meshes with anisotropic tetrahedral mesh extrusion in boundary layers (with optional recombination of tetrahedra)
- With community effort, we got there in approx. 6 months: special discretisation

Performance and Parallel Scaling Improvements

- Tuning of OpenFOAM for high-end parallel machines: collaboration with Intel
- Scalable Software Workshop: National Science Foundation, USA
- OpenMP baseline by Sandeep Menon, UMass Amherst
  - Basic wrapping and examples for use of OpenMP in the library
  - Objective: **provide multi-core support** at linear solver, matrix and calculus and field operator level, in stages!
- OpenFOAM on GPU in progress, with significant interest: more news coming soon

Massively Parallel Scaling



Intel Xeon Processor: X5650 B0; Frequency: 2.93 GHz # processors 2; # cores 6; Cache (L1 KB/L2 MB/L3 MB): 32 KB / 256 KB / 12 MB; 6x4GB DDR3 1333 RDIMM QDR InfiniBand 4x

VulaSHAKA Project

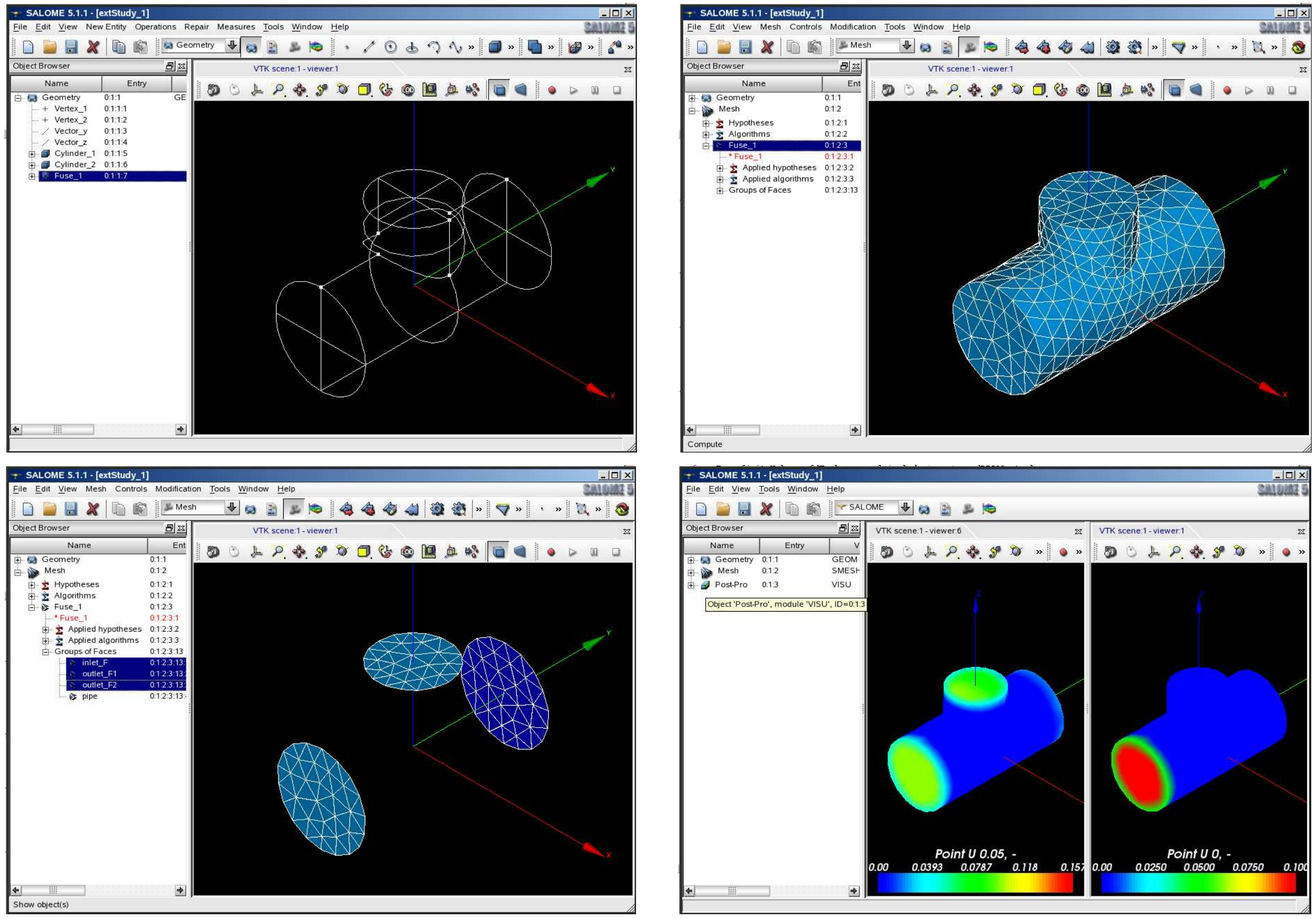
VulaSHAKA Background and Project Road-Map: OpenFOAM/SALOME Integration

- SALOME is an open source integration platform for numerical simulation
  - Development coordinated and financed by EDF (France)
  - Designed specifically to “host” numerical simulation software
- OpenFOAM/SALOME Integration
  - `IFoam`: Interactive FOAM, Python-based, explicit and implicit distributed process coupling and data translation
  - `GFoam`: Stand-alone GUI, including IFoam engine
  - `SFoam`: IFoam and GFoam embedded in a SALOME module

VulaSHAKA Project

VulaSHAKA Project Snapshots

- CAD generation, meshing, boundary condition setup and post-processing



VulaSHAKA Project

VulaSHAKA Code Components

- `IFoam` (Interactive FOAM): Interactive calculation and integration framework for OpenFOAM, provides integration to SALOME
- `pyFoam` (sorry, Bernhard): Python front-end to OpenFOAM, supports `IFoam`
- `unv2foam`: Extends OpenFOAM `ideasUnvToFoam` utility, introducing embedded capabilities necessary for embedding (available as C++ function)
- `foam2vtk`: Memory based conversion of FOAM objects into VTK objects for display
- `foam2med`: Allows translation of OpenFOAM data into MED format to enable integration with SALOME
- `confFoam`: Common configuration package for OpenFOAM development based on automake tools
- User requirements and future plans: **establish international collaboration**
- Developer credits
  - Ivor Clifford, (currently Penn State University), formerly PBMR
  - **Alexey Petrov**, Johannes Odendaal, PBMR Pty, South Africa

Deployment of OpenFOAM

OpenFOAM Extend Deployment and Release Schedule

- Substantial community contributions and developments need a better deployment framework: help the users with new capability and share results of your work
- Improvements in many directions needed: mimic operation of a software company
- Community effort is the only way to address all needs: large amount of work!

Major Improvements in the Pipeline

1. Unified Linux, Mac OS-X and Windows version
2. Improved system of binary release
3. Testing and validation harness
4. Community portal and joint documentation effort
5. Aiming for quarterly or six-monthly release schedule

New Features and Work-in-Progress

New Features

- Major new features
  - Finite area and thin liquid solver release: final version
  - Block matrix implementation with parallelisation support
  - Re-meshing with tetrahedral edge swapping: Sandeep Menon
  - Completion of the internal combustion engine library: Poly Milano
  - Parallelisation work in Fluid-Structure interaction solver
  - Python-SWIG interface to `OpenFOAM` and `finiteVolume` library: sufficient to migrate complete physics solvers into python
- Running OpenFOAM without trace on disk: complete functional case generation and setup. This is a part of python interface requirement – enable full interpreted execution of OpenFOAM cases, from mesh generation to post-processing
- Parallelisation of topology modifiers: parallel dynamic mesh (completed)
- Accuracy improvements on tetrahedral meshes
- Bug fixes and algorithmic improvements (approx 5000)
- New software management system to `git`

New Features and Work-in-Progress

Native Microsoft Windows Version of OpenFOAM

- Major piece of work, actively supported by Microsoft: HPC platform push
- No shortcuts: native compiler and operating system interface (non-POSIX)
- Complete rewrite of build system: `CMake`: generate Visual Studio project files
- New mechanism for include file handling: Michael Wild, FreeFOAM
- Library symbol import-export handling: changes to all class files!
- Plan: keep Windows port as a `git` branch to stabilise the regular release and resolve deployment and efficiency issues
- The work is lost if development lines do not merge. Therefore . . .

Unified Linux, Mac OS-X and Windows Version

- Merging the development line and regular porting required
- Binary releases for Linux, Mac and Windows; USB stick, Debian packages
- Public testing and validation runs prior to release
- Aiming for Quarterly or 6-monthly release schedule
- Planning a Release Committee to oversee and synchronise work and testing

Some Development Plans

Development Plans: Funded or Partially Funded Work

- Improvements in **parallel topological change library**: specifically, internal combustion engine simulation on 100s of processors
- Further work on **immersed boundary method**: looking for a complete physics-agnostic implementation with support for parallelism and dynamic mesh handling; improvements in execution speed and accuracy
- **Mixing plane interface**, in collaboration with industrial partners (Hydro Quebec): completing the tool-set for turbomachinery simulations
- Major push in **compressible turbomachinery** and strongly coupled density-based solvers with **geometric multigrid** acceleration
- Planning work on **harmonic balance solvers** in compressible turbomachinery arena (further funding is required)
- Work on **continuous and discrete adjoint solvers** for gradient-based optimisation needs
- Structural analysis and dynamics: simulation of stress waves and impact stresses; improvements in performance with the use of block-matrix solution techniques
- Completion of the **Python-Swig interface** for larger parts of the library: full interactive execution of all top-level solvers and utilities

Review

A Year in Life of OpenFOAM

- Very successful: improved capability, visibility and quality of the code
- This is a community-driven project with numerous contributors
- Presence of OpenFOAM in the numerical simulation arena is changing the way users are applying CFD in industrial settings
- Academic sharing of results and joint research is easier and more productive

Outlook

- Need to make OpenFOAM easier for entry-level users: several ongoing projects
- Formalise and validate software capability: adding new features is as important as making the best possible use of existing capability
- Expand the pool of expert developers: NUMAP-FOAM Summer School, Zagreb
- Grow the community and user base: software lives only as long as it is used
- **Sixth OpenFOAM Workshop**: Penn State University 13-16 June 2011 `http://www.openfoamworkshop.org`

. . . let's see what the future brings