OpenFOAM at FM– LTH

Erdzan Hodzic

Division of Fluid Mechanics, Department of Energy Sciences, Lund University

This offering is not approved or endorsed by ESI® Group, ESI-OpenCFD® or the OpenFOAM® Foundation, the producer of the OpenFOAM® software and owner of the OpenFOAM® trademark.
Outline

• History and Overview of OF
• OF at Fluid Mechanics LTH
• Why OF
• Cluster usage
• Best Practice
• Proposal for upcoming OF seminars
• Conclusions
History and Overview of OF:

Foam: Imperial College 1980s: H. Weller & H. Jasak

OpenFOAM 1.0

e.g. OpenFOAM-3.0.1

H. Weller = OpenCFD & ESI

H. Jasak = Wikki.ltd & OpenFOAM-extend
Why OpenFOAM:

- Expressive and versatile syntax
- Extensive capabilities
- Open architecture and open source (open source development)
- Object orientation:
  - Data Encapsulation: Grouping the data and functions together, e.g. sparse matrix class
  - Operator Overloading: high-level functionality, i.e. identical function names with different arguments
  - Object families and Run-time selection
  - Generic Programming: Independence of actual data type
Why not-OpenFOAM:

• Little official support

• Little coherent training material, many courses/presentations/reports/forum-posts needed for an answer.

• Overwhelming number of controllable parameters (in the beginning)
OF at Fluid Mechanics LTH: Which fork/version/variant

/home/

/srobi/

/soft/

/opt/

/moi

/openFOAM

foam

Root password: ******
OF at Fluid Mechanics LTH: Which fork/version/variant

OpenFOAM extensions

Looking for the latest version? Download foam-extend-3.1.sparseimage.zip (136.7 MB)

Source Pack Installation

Version 3.0.1, released 15/12/15

Packs
The following tar-zipped tgz source packs are available for download.

<table>
<thead>
<tr>
<th>Pack</th>
<th>File</th>
<th>md5sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenFOAM</td>
<td>OpenFOAM-3.0.1.tgz</td>
<td>304e5a14b9e69c20989527f5fb1ed724</td>
</tr>
<tr>
<td>Third-Party</td>
<td>ThirdParty-3.0.1.tgz</td>
<td>4665072d7d29ab9af5ced402f667185a</td>
</tr>
</tbody>
</table>
OF at Fluid Mechanics LTH: Which fork/version/variant
OF at Fluid Mechanics LTH: Which fork/version/variant

```
alias of221='source /opt/openfoam221/etc/bashrc'
alias of230='source $HOME/OpenFOAM/OpenFOAM-2.3.0/etc/bashrc'
alias of211='source $HOME/OpenFOAM/OpenFOAM-2.1.1/etc/bashrc'
alias of30ext='source $HOME/foam/foam-extend-3.0/etc/bashrc'
alias of20x='source $HOME/OpenFOAM/OpenFOAM-2.0.x/etc/bashrc'
alias OF20x='source $HOME/OpenFOAM/OpenFOAM-2.0.x/etc/myBashrc'
alias of240='source /opt/openfoam240/etc/bashrc'
alias of30x='source $HOME/OpenFOAM/OpenFOAM-3.0.x/etc/bashrc'
alias of301='source $HOME/OpenFOAM/OpenFOAM-3.0.1/etc/bashrc'
alias of32ext='source $HOME/foam/foam-extend-3.2/etc/bashrc'
```

```
endzan-3.0.1 endzan-3.0.x MyChemkinInputs OpenFOAM-2.0.x OpenFOAM-2.1.1
OpenFOAM-2.3.0 OpenFOAM-3.0.1 OpenFOAM-3.0.x ThirdParty-2.0.x ThirdParty-2.1.1
ThirdParty-2.3.0 ThirdParty-3.0.1 ThirdParty-3.6.x
```
OF at Fluid Mechanics LTH: Clusters

• High level representation of physical level ➔ parallelization, tensor algebra and mesh manipulation= automatically integrated (hidden) common for all solvers.

• HPC performance not always addressed, cluster/case dependent

• OF parallelization performed by MPI ➔ whole communication system in one single library. (OpenMPI)

• Well suited for wide range of MPI-versions, hardware and OS.

• Still, not well understood scalability and efficiency (ongoing debate)

• Several previous studies= 1e6-22e6 cells ➔ linear scalability up to 1024 cores
OF at Fluid Mechanics LTH: Clusters

- **System/decomposeParDict**

- **simple**: simple geometric decomposition in which the domain is split into pieces by direction, e.g. 2 pieces in the x direction, 1 in y etc.

- **hierarchical**: Hierarchical geometric decomposition which is the same as simple except the user specifies the order in which the directional split is done, e.g. first in the y-direction, then the x-direction etc.

- **metis**: METIS decomposition which requires no geometric input from the user and attempts to minimize the number of processor boundaries. The user can specify a weighting for the decomposition between processors which can be useful on machines with differing performance between processors.

- **manual**: Manual decomposition, where the user directly specifies the allocation of each cell to a particular processor.
OF at Fluid Mechanics LTH: Clusters

- **System/decomposeParDict**

  - **simple**: simple geometric decomposition in which the domain is split into pieces by direction, e.g. 2 pieces in the \( x \) direction, 1 in \( y \) etc.
  - **hierarchical**: Hierarchical geometric decomposition which is the same as simple except the user specifies the order in which the directional split is done, e.g. first in the \( y \)-direction, then the \( x \)-direction etc.
  - **metis**: METIS decomposition which requires no geometric input from the user and attempts to minimize the number of processor boundaries. The user can specify a weighting for the decomposition between processors which can be useful on machines with differing performance between processors.
  - **manual**: Manual decomposition, where the user directly specifies the allocation of each cell to a particular processor.

T. Lucchini, Running OpenFOAM in parallel
OF at Fluid Mechanics LTH: Clusters

• Example: pisoFOAM, 2.15e6 cells, Metis decomposition
OF at Fluid Mechanics LTH: Clusters

- Results:
  - GAMG vs BiCG
  - bottlenecks
OF at Fluid Mechanics LTH: Clusters

• Other studies same geom:
  – No diff. Intel vs GNU
  – Regardless of OF vers.

M. Culpo, Current Bottlenecks in the scalability of OpenFOAM on Massively Parallel Clusters, www.prace-ri.eu
OF at Fluid Mechanics LTH: Clusters

• Other studies, dieselFOAM

<table>
<thead>
<tr>
<th>Object name</th>
<th>Time consuming(~)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UEqn</td>
<td>2.8s</td>
</tr>
<tr>
<td>pEqn × 2</td>
<td>6s</td>
</tr>
<tr>
<td>hEqn</td>
<td>2.35s</td>
</tr>
<tr>
<td>YEqn</td>
<td>2.47s</td>
</tr>
<tr>
<td>Turbulence (k, epsilon)</td>
<td>2.51s</td>
</tr>
<tr>
<td>rhoEqn</td>
<td>1.3s</td>
</tr>
<tr>
<td>etc.</td>
<td>3.4s</td>
</tr>
</tbody>
</table>

Local-test: using MPI_* commands

Y. Liu, Hybrid Parallel Computation of OpenFOAM Solver on Multi-Core Cluster Systems
OF at Fluid Mechanics LTH:

How to use OpenFOAM on PDC machines

Openfoam is a free, open source CFD software package. See also [http://www.openfoam.org](http://www.openfoam.org).

To see which versions of openfoam are installed on any of the machines at PDC log into the machine and type

```
module avail openfoam
```

You should consider questions before running openFOAM such as:

- How often do you save your solution? What trace/history of your iterations do you write to file(s)?

To control this behavior you need to modify the corresponding parameters in

```
controlDict
```

Moreover you can adapt your simulation to Beskow by tweaking the following parameters:

```
writeCompression
runTimeModifiable
```

uncompressed;
no;
**OF best practise:**

<table>
<thead>
<tr>
<th>Write settings</th>
<th>Used space</th>
<th>reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascii</td>
<td>45.5 MB</td>
<td></td>
</tr>
<tr>
<td>ascii, compressed</td>
<td>16.7 MB</td>
<td>28.8 MB</td>
</tr>
<tr>
<td>binary</td>
<td>33.8 MB</td>
<td></td>
</tr>
<tr>
<td>binary, compressed</td>
<td>28.8 MB</td>
<td>16.7 MB</td>
</tr>
</tbody>
</table>


1. **binary**: 0.142s
2. **ascii_sprintf_g**: 2.609s
3. **binary compressed**: 2.831s
4. **ascii**: 3.102s
5. **ascii_sprintf_g compressed**: 3.67s
6. **ascii compressed**: 4.164s

**Binary Uncompressed** - Fastest

**ASCII Compressed** - ~25% Slower

**ASCII Uncompressed** - ~40% Slower

**Binary Compressed** - ~115% Slower

https://github.com/wyldckat/wyldckat.github.io/wiki/OF23_IOPerformance_Analysis_2

OF best practise: GPU and hyper-threading
OF best practise: GPU and hyper-threading

Increased overhead will outweigh any benefit gained from maximizing CPU utilization through the use of virtual cores.
OF best practise: GPU and hyper-threading

..increased overhead will outweigh any benefit gained from maximizing CPU utilization through the use of virtual cores.
OF at Fluid Mechanics LTH: Clusters

- Other studies at FM, P. Iudiciani, M. Selewski etc.
  - ~100,000 cells/cpu
Cluster usage: Possible improvements

Usage per day for project SNIC 2015/16-13 on Aurora @ Lunarc

SNIC 2015/16-25 on Beskow @ PDC

SNIC 2015/16-25 on Abisko @ HPC2N
Proposal for FM-User exchange

• Jonny/4FM/
  – fieldToVector & vectorToField
  – outputMesh & outputVolume
  – MyChemkinInputs = All CHEMKIN *.inp/*.therm/*.tran
  – postAverage-org = use OF libSampling utilities
Proposal for upcoming OF subjects

• Five Main Objects of OF
• BC:s => best practice
• Numerical Schemes => best practice
• Spray Modelling in OF
• Dynamic meshes in OF
• Combustion Models in OF
Useful links:

- The OpenFOAM User Guide: http://www.openfoam.org/docs/user/
- The CFD Online Forum: http://www.cfd-online.com/Forums/openfoam/
- The OpenFOAM Wiki: http://openfoamwiki.net/index.php/Main_Page
- The CoCoons Project: http://www.cocoons-project.org/
- Chalmers University OF course: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD/
- 101 Things to read when starting with OF: http://www.sourceflux.de/blog/101-things-read-starting-openfoam/
Summary:

- Local installation pf OF-forks
- 50.000-150.000 cells per cpu
- Use Metis or Scotch
- Save data in binary/uncompressed
- Wait with GPU
- Read: ”101Things to read when starting with OF”
Thank you for your attention!