FULL-SCALE SIMULATION OF AN INDUSTRIAL STEAM CRACKING FURNACE USING OPENFOAM ON TIER-1

Pieter A. Reyniers, Laurien A. Vandewalle, Kevin M. Van Geem, Guy B. Marin

Laboratory for Chemical Technology (LCT) – Ghent University
laurien.vandewalle@ugent.be
pieter.reyniers@ugent.be
MULTISCALE MODELING APPROACH

Process

Separation units

Reactor

Transport phenomena

Materials design

Kinetics

Surface phenomena

Mechanism

Ab initio

Fluid phase

Zeolite

Metal sites

Acid sites

Physisorption

(d-e)-hydrogenation

(d-e)-protonation

alkyl-shift

PCP-branching

ß-scission

13/09/2017 2/21
STEAM CRACKING

Crude oil

Steam cracking

Consumer goods from chemical industry

Cracked gas

Stack

Cracked feed

Stack

Radiant section

Burners

Process gas

Fuel (natural gas)

Natural gas

Bio-based feeds
COKE FORMATION IN STEAM CRACKING

Endothermic process at temperatures of 800–900 °C
Deposition of a carbon layer on the reactor surface
- Reduced thermal efficiency
- High pressure causes loss of product selectivity
- Coil carburization and thermal stress

Coke reduction method: 3D reactor technology

\[
r_C = \sum_i c_i \cdot A_i \cdot \exp\left(\frac{-E_{a,i}}{RT_{int}}\right)
\]

Coil cracking due to differences in thermal expansion rate

Hot spots due to inhomogeneous coke formation

Nova Chemicals, 2002; Linde Group; Muños et al., 2013; Albright et al., 1988; Muños et al., 2014
<table>
<thead>
<tr>
<th></th>
<th>KBR / S+C</th>
<th>Kubota (Slit- / X-) MERT®</th>
<th>Lummus/Sinopec IHT®</th>
<th>Technip Swirl Flow Tube®</th>
</tr>
</thead>
</table>

(*) Van Cauwenberge et al. 2015, Chem. Eng. J.
DYNAMIC RUN LENGTH SIMULATION

- Run simulation for $t_{sim}$ time steps
- $t_i = t_{i-1} + t_{sim}$
- $\Delta T_{MT} \geq \Delta T_{MT_{max}}$ → YES → End
- $\Delta p \geq \Delta p_{max}$ → NO
- $t_i = t_{i-1} + 1$
- Coke layer growth
- Mesh update

Propane Millisecond
29 species, 151 reactions

Axial position [m]

Heat flux [W m$^{-2}$]

Increasing run length

SOR (0 hrs)

SOR (48 hrs, 96 hrs)

Pressure drop [Pa]

Run length [days]

Time on stream [days]

$C_2H_4$ selectivity [-]

## OPENFOAM ON TIER-1

### Hardware overview

<table>
<thead>
<tr>
<th></th>
<th>Tier-1b – BrENIAC</th>
<th>Tier-1a – Muk</th>
<th>Tier-2 – Golett</th>
<th>Tier-2 – Swalot</th>
</tr>
</thead>
<tbody>
<tr>
<td># nodes</td>
<td>580</td>
<td>528</td>
<td>200</td>
<td>128</td>
</tr>
<tr>
<td>CPU</td>
<td>2 x 14c Intel E5-2680v4</td>
<td>2 x 8c Intel E5-2670</td>
<td>2 x 12c Intel E5-2680v3</td>
<td>2 x 10c Intel E5-2660v3</td>
</tr>
<tr>
<td>Memory</td>
<td>128 GiB (435)</td>
<td>64 GiB</td>
<td>64 GiB</td>
<td>128 GiB</td>
</tr>
<tr>
<td></td>
<td>256 GiB (145)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interconnect</td>
<td>EDR IB (11.75 GB/s)</td>
<td>FDR IB (6.5 GB/s)</td>
<td>FDR-10 IB (5.0 GB/s)</td>
<td>FDR IB (6.5 GB/s)</td>
</tr>
<tr>
<td>Access</td>
<td>Project-based, free</td>
<td>Project-based, paying/free</td>
<td>Open, free</td>
<td>Open, free</td>
</tr>
</tbody>
</table>
OPENFOAM ON TIER-1

Information on project access

- Project proposal in a single document (maximum 17 pages)
- Scientific relevance is demonstrated by framing the calculation time in an approved project
- Next cut-off date for proposals: October 2, 2017.
- Possibility of requesting a starting grant (continuous call)
- FWO bears all the cost but the number of nodedays is limited

- Nearly identical in use compared to UGent Tier-2 machines (modules, scheduler, job-scripts)
- Major difference: accounting system to keep track of consumed nodedays
- Connection between BrENIAC (@KULeuven) and UGent via BelNet (1 Gbps).
OPENFOAM ON TIER-1

Scaling on Tier-1b

Better scaling compared to Tier-1a – Muk

Fast interconnect (EDR IB) reduces wall-clock time and maintains efficiency while scaling on more cores

Bottleneck: pre- and postprocessing

Remote desktop on Tier-1b login node with GPU via NoMachine client.
Super-linear scalability due to cache effect and better accommodation of memory patterns across multiple nodes

The choice of \textit{decomposition method} (scotch, simple, metis, etc.) is important for the number of processor faces

Use preservePatches for periodic cases
IMPROVED SCALING FOR OPENFOAM

Meshers generated with native meshers (blockMesh, snappyHexMesh) calculate quicker than third-party meshes.

Use `renumberMesh` to decrease bandwidth and increase speed for third-party meshes.

Use OpenFOAM versions compiled with recent compiler toolchains.

```
---------------------------------- /apps/leuven/broadwell/2016a/modules/all ----------------------------------
OpenFOAM/2.2.0-intel-2016a       OpenFOAM/2.3.1-intel-2016a       OpenFOAM/4.1-intel-2016a
OpenFOAM/2.2.2-intel-2016a       OpenFOAM/3.0.1-intel-2016a       OpenFOAM/Extend/3.2-intel-2016a
OpenFOAM/2.3.1-intel-2017a       OpenFOAM/3.0.1-intel-2016b
---------------------------------- /apps/gent/SL6/sandybridge/modules/all ----------------------------------
OpenFOAM-Extend/3.2-intel-2016a   OpenFOAM/2.3.1-intel-2015a
OpenFOAM/2.1.1-ictce-4.0.10      OpenFOAM/2.4.0-intel-2015b
OpenFOAM/2.2.0-ictce-4.1.13      OpenFOAM/3.0.0-intel-2015b-eb-deps-Python-2.7.10
OpenFOAM/2.2.2-intel-2015a      OpenFOAM/3.0.1-intel-2016b
OpenFOAM/2.2.2-intel-2016a       OpenFOAM/4.0-intel-2016b
OpenFOAM/2.3.0-intel-2014b       OpenFOAM/Extend/3.2-intel-2016a
```

Tier-1b
Use **checkpointing** at reasonable time intervals, consider **file compression** for large cases

```plaintext
writeControl timeStep;
writeInterval 1000;
purgeWrite 2;
writeFormat ascii;
writePrecision 6;
writeCompression on;
```

Writing out every 2-3 hours for a 72h job is sufficient

Only keep the last # time steps, earlier time steps are removed

For large cases, consider to write out the files in .gz format
IMPROVED SCALING FOR OPENFOAM

Run your job from the appropriate location, excessive I/O on low-bandwidth locations will seriously slow down your job

$VSC\_DATA$: not meant for calculations, only long-term storage

$VSC\_SCRATCH$: default scratch on 15k disks

$VSC\_SCRATCH\_NODE$: /tmp location on local node, only accessible as long as the jobs is running, suited for single-node jobs

Disable runTimeModifiable in controlDict to avoid excessive stat() calls at every time step

```
runTimeModifiable false;
```

IMPROVED SCALING FOR OPENFOAM

NEW in OpenFOAM v5.0: collated file format

“the data for each decomposed field (and mesh) is collated into a single file that is written (and read) on the master processor. The files are stored in a single directory named processors.”

“The file writing can be threaded allowing the simulation to continue running while the data is being written to file.”

IMPROVED SCALING FOR OPENFOAM

CASE STUDY (OpenFOAM 1.7)

Metadata handling becomes a bottleneck when scaling on a large number of cores due to an increasing volume of small files

<table>
<thead>
<tr>
<th>Number of processes</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute time [s]</td>
<td>686</td>
<td>801</td>
<td>890</td>
<td>1161</td>
<td>2248</td>
</tr>
<tr>
<td>Cumulative metadata [s]</td>
<td>64</td>
<td>202</td>
<td>274</td>
<td>389</td>
<td>892</td>
</tr>
<tr>
<td>Metadata share [%]</td>
<td>9%</td>
<td>25%</td>
<td>31%</td>
<td>34%</td>
<td>39%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of processes</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td># files created</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
<td>4096</td>
<td>8192</td>
</tr>
<tr>
<td># files read</td>
<td>1089</td>
<td>2177</td>
<td>4353</td>
<td>9729</td>
<td>17409</td>
</tr>
<tr>
<td>Average file size</td>
<td>597K</td>
<td>317K</td>
<td>163K</td>
<td>84K</td>
<td>47K</td>
</tr>
<tr>
<td># stat() calls</td>
<td>500,000</td>
<td>1,000,000</td>
<td>2,000,000</td>
<td>4,400,000</td>
<td>8,500,000</td>
</tr>
</tbody>
</table>

Curie
- owned by GENCI
- operated into the TGCC by CEA

- B510 bullx nodes
- 2 x 8c Intel E5-2680
- 64 GiB
- local SSD disk
- QDR IB Full Fat Tree
- LUSTRE storage (150 GB/s)


IMPROVED SCALING FOR OPENFOAM

CASE STUDY (OpenFOAM 1.7)

stat()-calls are use to check the timestamps of files to check for updates. Disabling ‘runTimeModifiable’ reduces the number of stat()-calls drastically.

<table>
<thead>
<tr>
<th>Number of processes</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute time [s]</td>
<td>542</td>
<td>381</td>
<td>343</td>
<td>411</td>
</tr>
<tr>
<td>Cumulative metadata [s]</td>
<td>0.99</td>
<td>2.62</td>
<td>6.03</td>
<td>14.4</td>
</tr>
<tr>
<td>Metadata share [%]</td>
<td>0.20%</td>
<td>0.70%</td>
<td>1.80%</td>
<td>3.50%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of processes</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
</tr>
</thead>
<tbody>
<tr>
<td># files created</td>
<td>512</td>
<td>1024</td>
<td>2048</td>
<td>4096</td>
</tr>
<tr>
<td># files read</td>
<td>1089</td>
<td>2177</td>
<td>4353</td>
<td>9729</td>
</tr>
<tr>
<td>Average file size</td>
<td>597K</td>
<td>317K</td>
<td>163K</td>
<td>84K</td>
</tr>
<tr>
<td># stat() calls</td>
<td>5,000</td>
<td>10,000</td>
<td>20,000</td>
<td>44,000</td>
</tr>
</tbody>
</table>

Curie
owned by GENCI
operated into the TGCC by CEA

B510 bullx nodes
2 x 8c Intel E5-2680
64 GiB
local SSD disk
QDR IB Full Fat Tree
LUSTRE storage (150 GB/s)

A project by Ghent University which aims to develop a flexible, open source Large-Eddy Simulations (LES) Computational Fluid Dynamics (CFD) code-base for multiscale modelling of several multidisciplinary applications.

Objectives defined in the following fields
1. Reduced chemistry
2. Sprays
3. Turbulent steady spray flames
4. Unsteady sprays, in internal combustion engines
5. Fire dynamics
ACKNOWLEDGEMENTS

Colleagues @LCT

David Van Cauwenberge
Pieter Reyniers
Laurien Vandewalle
Pieter Verhees
Jens Dedeyne
Moreno Geerts
Shekhar Kulkarni
Yu Zhang
LABORATORY FOR CHEMICAL TECHNOLOGY

Tech Lane Ghent Science Park – Campus A
Technologiepark 914, 9052 Ghent, Belgium

E info.lct@ugent.be
T 0032 9 331 17 57

https://www.lct.ugent.be