Running a 101 million cell case in OPM Flow

and the impact of ACROSS on OPM

Kjetil Olsen Lye (SINTEF Digital)

OPM Summit 2024

The simulation

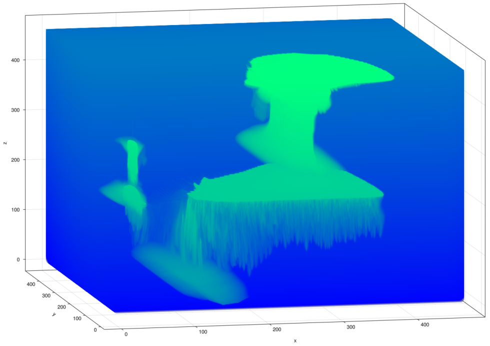
We used this:

32 x 2 x AMD 7H12 (64 cores, 2.6 GHz)

256 GB RAM / node.

(The cluster has 720 CPU nodes)

to make OPM Flow compute this:

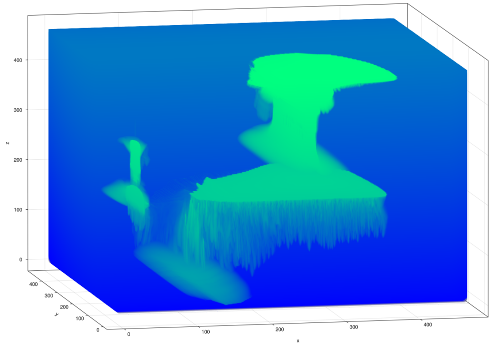


Video by Olav Møyner @ SINTEF Digital

The simulation

We used this:

to make OPM Flow compute this:



Video by Olav Møyner @ SINTEF Digital

The end?

The ACROSS EU Project

HPC, Big Data, and Artificial Intelligence convergent platform [ACROSS]

- 2021 – 2024
- Workflow centric use of ML and traditional HPC
- SINTEF's role: simulation of CO_2 storage pilot

ACROSS: In-situ processing support in OPM Flow

OPM OPEN POROUS MEDIA

Damaris has been integrated into OPM Flow, supporting:

- parallel HDF5 output
- in-situ remote ParaView visualization
- Python processing with DASK support

ACROSS: In-situ processing support in OPM Flow

Damaris has been integrated into OPM Flow, supporting:

- parallel HDF5 output
- in-situ remote ParaView visualization
- Python processing with DASK support

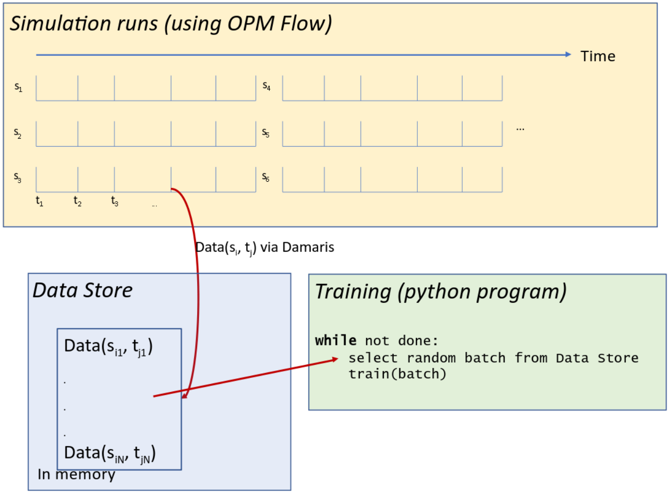
ACROSS: In-situ processing support in OPM Flow

Damaris has been integrated into OPM Flow, supporting:

- parallel HDF5 output
- in-situ remote ParaView visualization
- Python processing with DASK support

```
def main(DD):
    # (...)
    pressure = DD['iteration_data']['PRESSURE']['numpy_data']['P0_B0']
    np.savetxt("pressure.txt", pressure)
```

Consequence of Damaris: In-situ learning support



```
while not done:
    select random batch from Data Store
    train(batch)
```

KPI in ACROSS:

Simulate 100M cells for 1000 years.

The Hardware: The Karolina Cluster at IT4I

CPU partition:

- 720x 2x AMD 7H12
- 64 cores/CPU, 2.6 GHz
- 92,160 cores in total
- 256 GB RAM / node

GPU partition:

- 72x 2x AMD 7763
- 64 cores/CPU, 2.45 GHz
- 9,216 cores in total
- 72x 8x NVIDIA A100 GPU
- 576 GPUs in total
- 1024 GB RAM / node

The Hardware: The Karolina Cluster at IT4I

CPU partition:

- 720x 2x AMD 7H12
- 64 cores/CPU, 2.6 GHz
- 92,160 cores in total
- 256 GB RAM / node

GPU partition:

- 72x 2x AMD 7763
- 64 cores/CPU, 2.45 GHz
- 9,216 cores in total
- 72x 8x NVIDIA A100 GPU
- 576 GPUs in total
- 1024 GB RAM / node

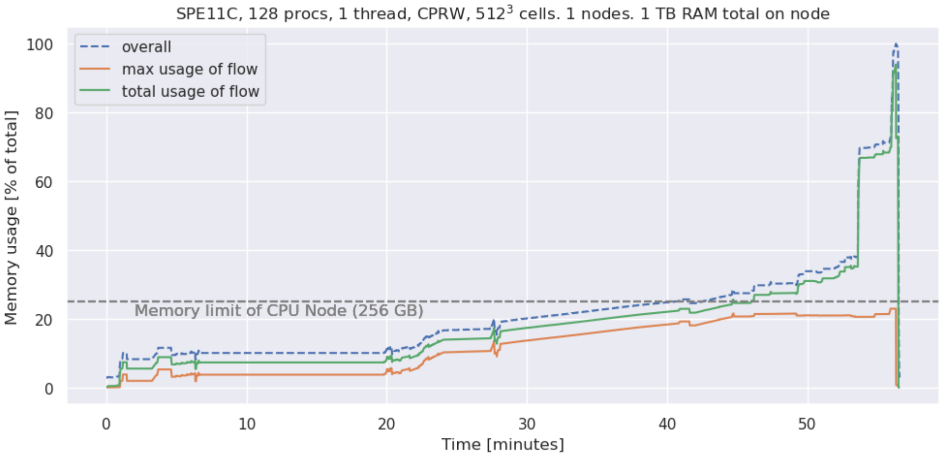
The case: SPE11C

We use the SPE11C case with

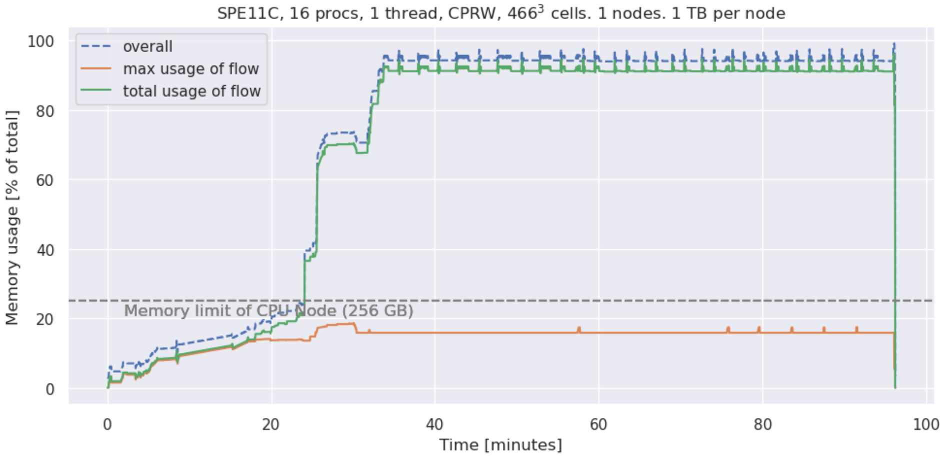
- Cartesian, $468 \times 466 \times 466$
- First 1000 years equilibrium disabled
- Thermal disabled

Problem: Only 256 GB RAM / node.

Memory usage of OPM Flow as a function of time



Memory usage of without thermal



Problem: Only 256 GB RAM/node. ✓
Solution: Disable thermal, run on 32 nodes.

Problem: Only 256 GB RAM/node. ✓

Problem: Zoltan crashes.

Partitioning issues with Zoltan

Zoltan crashes at around 32 million Cartesian cells

- Unweighted graph
- Both serial and parallel
- 64-bits support enabled
- Reproducible outside of OPM Flow

However, METIS is able to partition the graph.

1. Run OPM Flow to dump graph to file
2. Partition said file with serial METIS[1]
3. Run OPM Flow with partition file from METIS

[1] Takes around 1 minute

Partitioning issues with Zoltan

Zoltan crashes at around 32 million Cartesian cells

- Unweighted graph
- Both serial and parallel
- 64-bits support enabled
- Reproducible outside of OPM Flow

However, METIS is able to partition the graph.

1. Run OPM Flow to dump graph to file
2. Partition said file with serial METIS[1]
3. Run OPM Flow with partition file from METIS

[1] Takes around 1 minute

Partitioning issues with Zoltan

Zoltan crashes at around 32 million Cartesian cells

- Unweighted graph
- Both serial and parallel
- 64-bits support enabled
- Reproducible outside of OPM Flow

However, METIS is able to partition the graph.

1. Run OPM Flow to dump graph to file
2. Partition said file with serial METIS[1]
3. Run OPM Flow with partition file from METIS

[1]Takes around 1 minute

Partitioning issues with Zoltan

Zoltan crashes at around 32 million Cartesian cells

- Unweighted graph
- Both serial and parallel
- 64-bits support enabled
- Reproducible outside of OPM Flow

However, METIS is able to partition the graph.

1. Run OPM Flow to dump graph to file
2. Partition said file with serial METIS[1]
3. Run OPM Flow with partition file from METIS

[1]Takes around 1 minute

Problem: Only 256 GB RAM/node. ✓

Problem: Zoltan crashes. ✓

Solution: Use METIS.

Problem: Only 256 GB RAM/node. ✓

Problem: Zoltan crashes. ✓

Problem: Segmentation fault.

A segmentation fault appears

```
Processing grid
[acn17:120476:0:120476] Caught signal 11 (Segmentation fault: address not
    mapped to object at address 0x10)
==== backtrace (tid: 120476) ====
 0 0x000000000383fb45 finduniquepoints()  ???:0
 1 0x000000000383dcd5 process_grdecl()  ???:0
 2 0x000000000380add2 Dune::cpgrid::CpGridData::processEclipseFormat()  ???:0
 3 0x000000000380ea41 Dune::cpgrid::CpGridData::processEclipseFormat()  ???:0
```

`int` is used liberally within OPM, and

$$\overbrace{101\,000\,000}^{\text{number of cells}} \cdot \underbrace{8}_{\text{sizeof(double)}} \cdot \overbrace{8}^{\text{number of corners}} = 6\,464\,000\,000 > \text{MAX\_INT}$$

Made an isolated test and fixed overflow in

- `opm/grid/cpgpreprocess/preprocess.c`
- `opm/grid/cpgpreprocess/unique.c`

A segmentation fault appears

```
Processing grid
[acn17:120476:0:120476] Caught signal 11 (Segmentation fault: address not
    mapped to object at address 0x10)
==== backtrace (tid: 120476) ====
 0 0x000000000383fb45 finduniquepoints()  ???:0
 1 0x0000000000383dcd5 process_grdecl()  ???:0
 2 0x00000000000380add2 Dune::cpgrid::CpGridData::processEclipseFormat()  ???:0
 3 0x00000000000380ea41 Dune::cpgrid::CpGridData::processEclipseFormat()  ???:0
```

`int` is used liberally within OPM, and

$$\overbrace{101\ 000\ 000}^{\text{number of cells}} \cdot \underbrace{8}_{\text{sizeof(double)}} \cdot \overbrace{8}^{\text{number of corners}} = 6\ 464\ 000\ 000 > \text{MAX\_INT}$$

Made an isolated test and fixed overflow in

- `opm/grid/cpgpreprocess/preprocess.c`
- `opm/grid/cpgpreprocess/unique.c`

A segmentation fault appears

```
Processing grid
[acn17:120476:0:120476] Caught signal 11 (Segmentation fault: address not
    mapped to object at address 0x10)
==== backtrace (tid: 120476) ====
 0 0x000000000383fb45 finduniquepoints()  ???:0
 1 0x000000000383dcd5 process_grdecl()  ???:0
 2 0x000000000380add2 Dune::cpgrid::CpGridData::processEclipseFormat()  ???:0
 3 0x000000000380ea41 Dune::cpgrid::CpGridData::processEclipseFormat()  ???:0
```

`int` is used liberally within OPM, and

$$\overbrace{101\,000\,000}^{\text{number of cells}} \cdot \underbrace{8}_{\text{sizeof(double)}} \cdot \overbrace{8}^{\text{number of corners}} = 6\,464\,000\,000 > \text{MAX\_INT}$$

Made an isolated test and fixed overflow in

- `opm/grid/cpgpreprocess/preprocess.c`
- `opm/grid/cpgpreprocess/unique.c`

Problem: Only 256 GB RAM/node. ✓

Problem: Zoltan crashes. ✓

Problem: Segmentation fault. ✓

Solution: `int` overflow fixes.

Run summary

```
================    End of simulation     ===============

Number of MPI processes:       1024
Threads per MPI process:          4
Number of timesteps:          10050
Total time (seconds):         78264.74
Solver time (seconds):        78242.29
 Assembly time (seconds):      9674.80 (Failed: 3.5; 0.0%)
   Well assembly (seconds):       0.00 (Failed: 0.0; 0.0%)
 Linear solve time (seconds):52800.33 (Failed: 43.8; 0.1%)
   Linear setup (seconds):     17624.18 (Failed: 13.3; 0.1%)
 Update time (seconds):        12932.66 (Failed: 4.6; 0.0%)
 Pre/post step (seconds):      2194.43 (Failed: 0.1; 0.0%)
 Output write time (seconds):  587.14
Overall Linearizations:       30377    (Failed:  11; 0.0%)
Overall Newton Iterations:    20330    (Failed:  11; 0.1%)
Overall Linear Iterations:   117635    (Failed:  80; 0.1%)
```

Problem: Only 256 GB RAM/node. ✓

Problem: Zoltan crashes. ✓

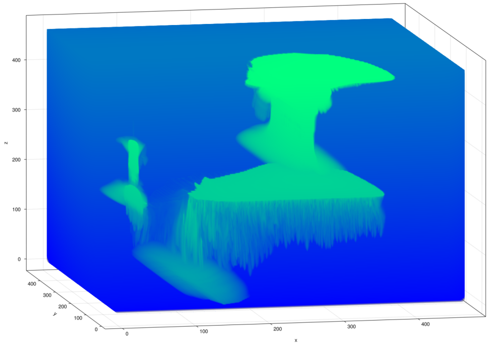
Problem: Segmentation fault.✓

Problem: Output files were corrupt.

Output problems

The workflow for visualizing the results:[2]

1. Load the file with MRST/MATLAB in debug mode
2. Dump the data to new, plain data files
3. Load said data files in Julia
4. Visualize with GLMakie.jl



[2]Done by Olav Møyner @ SINTEF Digital

Problem: Only 256 GB RAM/node. ✓

Problem: Zoltan crashes. ✓

Problem: Segmentation fault. ✓

Problem: Output files were corrupt. ✓

Solution: MRST/MATLAB reading

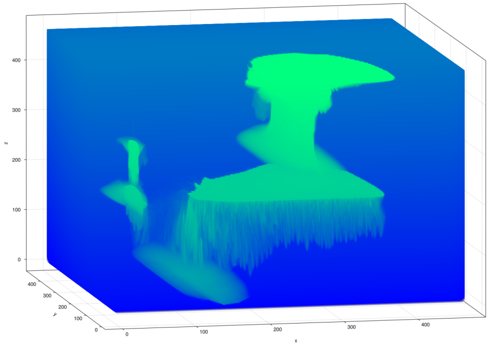
Conclusions and future work

Conclusions

- We ran a 101 million cell case (90 000 core hours)
- Running large cases in OPM Flow is not trivial

Future work to make running large cases trivial:

- replace `int` with `long long` or `size_t`
- support METIS partitioning
- partition in isolated process
- reduce peak memory usage
- isolate and fix output corruption.



Thank you.

Kjetil Olsen Lye

`kjetil.olsen.lye@sintef.no`