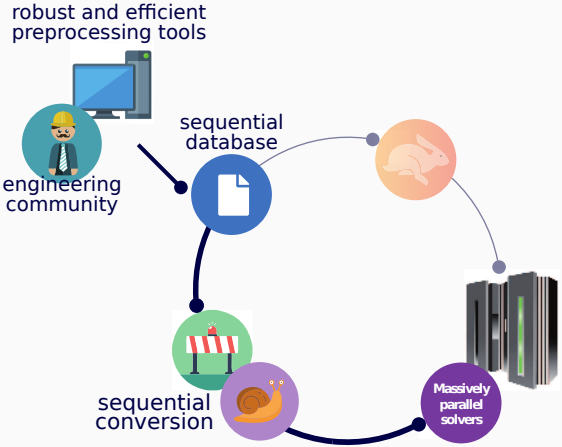
Workflow for Parallel Processing of Sequential Mesh Databases

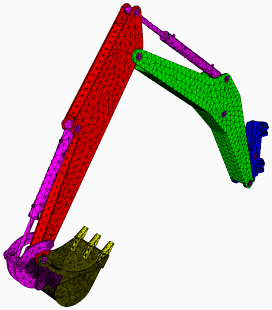
Ondřej Meca, Lubomír Říha, Tomáš Brzobohatý

IT4Innovations, National Supercomputing Center

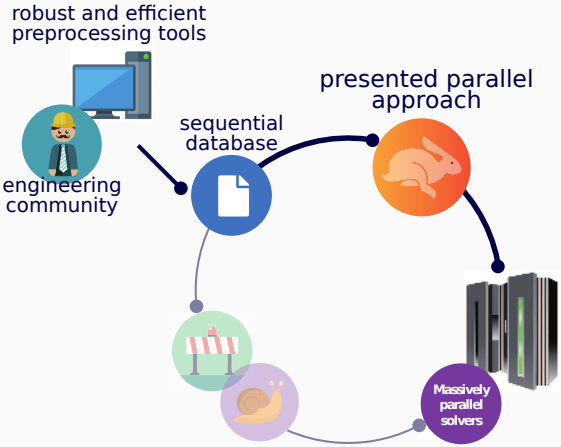
Motivation

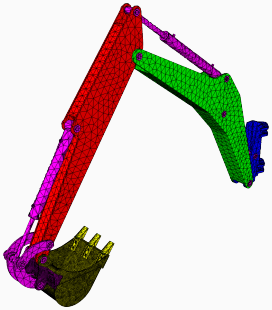
Motivation:
Simplify usage of HPC for the engineering community





Motivation:
Simplify usage of HPC for the engineering community



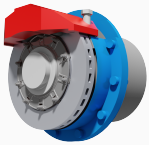


Processing of Model Databases

Model Databases – unstructured meshes

Valve, 6.3 GB / 31M / 23M

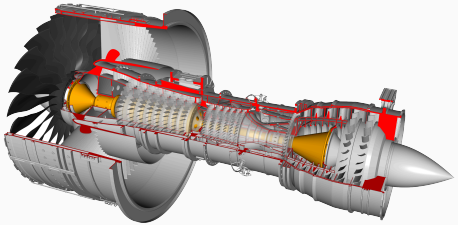
Manifold, 21 GB / 101M / 73M



Disc Brake, 19 GB / 122M / 45M

Loader Arm, 28 GB / 169M / 85M

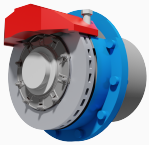
Jet Engine	
file size	142 GB
nodes	822 M
elements	484 M



Model Databases – unstructured meshes

Valve, 6.3 GB / 31M / 23M
433 s → **2.22 s**

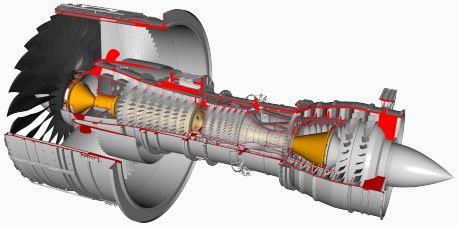
Manifold, 21 GB / 101M / 73M
1359 s → **3.77 s**



Disc Brake, 19 GB / 122M / 45M
938 s → **4.21 s**

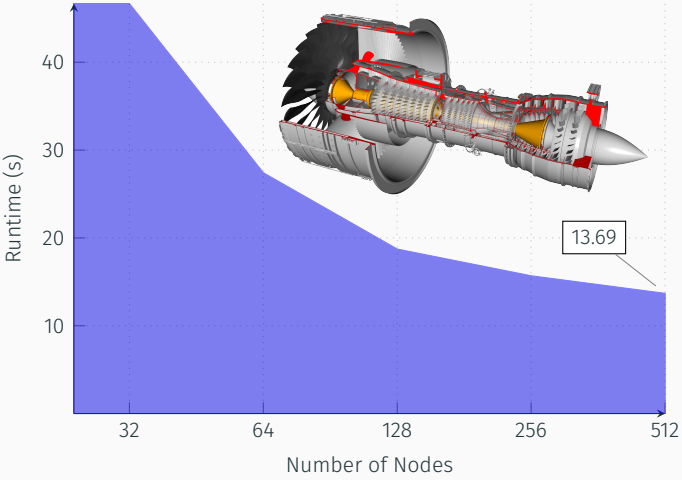
Loader Arm, 28 GB / 169M / 85M
1270 s → **4.87 s**

Jet Engine	
file size	142 GB
nodes	822 M
elements	484 M
seq. processing	∼*7200* s
par. processing	**13.69 s**

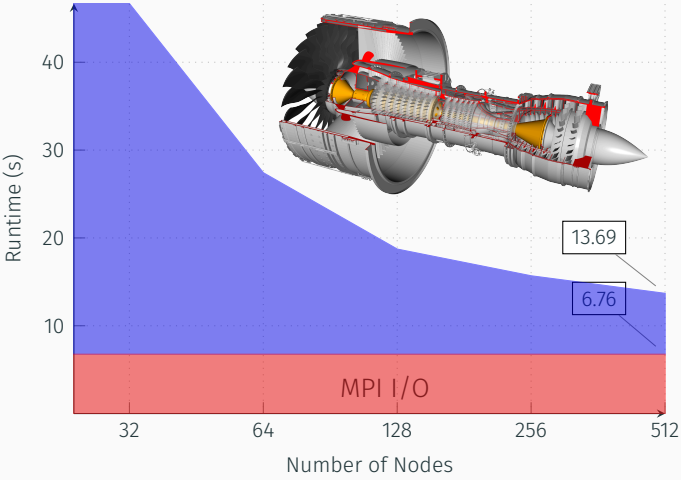


Workflow for Parallel Processing

Processing of the Jet Engine example up to 6144 MPI processes



Processing of the Jet Engine example up to 6144 MPI processes



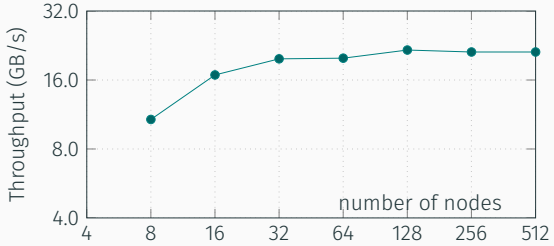
MPI I/O – from 500MB/s to 30GB/s

Divide a file among all object storages

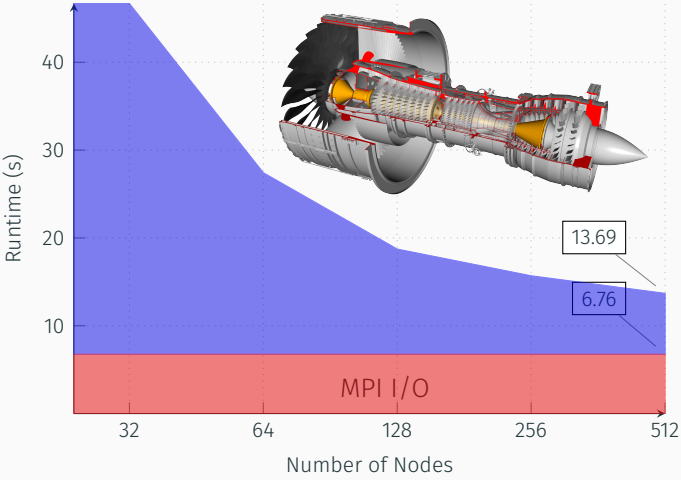
Use large stripes

Reduce the number of reading processes

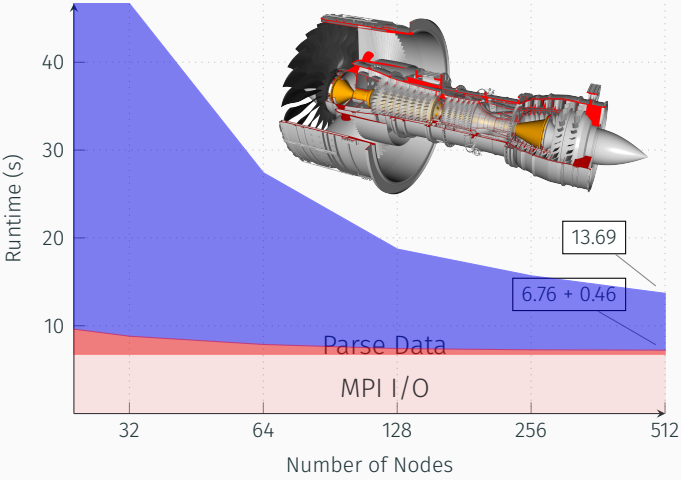
Reading up to 70% of the Salomon maximal throughput (30GB/s)



Processing of the Jet Engine example up to 6144 MPI processes

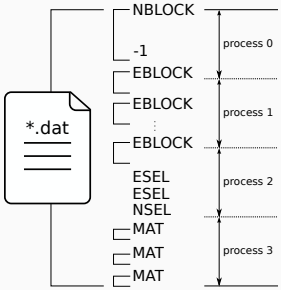


Processing of the Jet Engine example up to 6144 MPI processes



Parsing of ANSYS Workbench

A sequential mesh database (in ASCII format)



Data organized in blocks defined by commands

Processing of Parsed Data – Challenges

Nodes and elements are randomly scattered across MPI processes

- How to efficiently balance data among processes?
- How to get processes where coordinates should be sent?
- How to assign regions to nodes and elements?

```
- - - MPI 0 - - -
NODES
ID     X     Y     Z
 1 1.00 3.00 0.00
 0 0.00 3.00 0.00
 4 1.00 2.00 0.00
 2 2.00 3.00 0.00
 3 0.00 2.00 0.00
10 1.00 0.00 0.00
 5 2.00 2.00 0.00
```

```
- - - MPI 1 - - -
NODES
ID     X     Y     Z
 9 0.00 0.00 0.00
 7 1.00 1.00 0.00
 8 2.00 1.00 0.00
 6 0.00 1.00 0.00
11 2.00 0.00 0.00

ELEMENTS
ID TYPE NODES
 3    8    7  8  5  4
 0    8    3  4  1  0
```

```
- - - MPI 2 - - -
ELEMENTS
ID TYPE NODES
 1    8    4  5  2  1
 4    8    9 10  7  6
 5    8   10 11  8  7
 2    8    6  7  4  3
 6    2    5  2
 8    2   11  8
 7    2    8  5
```

```
- - - MPI 3 - - -
NODE REGION BLACK
0 1 3 6 9 10
ELEMENT REGION LEFT
0 2 4
ELEMENT REGION RIGHT
1 3 5
ELEMENT REGION FACE
6 7 8
```

Processing of Parsed Data – Parallel sorting

Rearrange nodes and elements to a known distribution by the **parallel histogram sort**

Assign regions to nodes and elements

Build the mesh and compute the final decomposition suitable for a parallel solver?

- - - MPI 0 - - -

NODES

0 0.00 3.00 0.00
1 1.00 3.00 0.00
2 2.00 3.00 0.00

ELEMENTS

0 8 3 4 1 0
1 8 4 5 2 1
2 8 6 7 4 3

- - - MPI 1 - - -

NODES

3 0.00 2.00 0.00
4 1.00 2.00 0.00
5 2.00 2.00 0.00

ELEMENTS

3 8 7 8 5 4
4 8 9 10 7 6
5 8 10 11 8 7

- - - MPI 2 - - -

NODES

6 0.00 1.00 0.00
7 1.00 1.00 0.00
8 2.00 1.00 0.00

ELEMENTS

6 2 5 2
7 2 8 5
8 2 11 8

- - - MPI 3 - - -

NODES

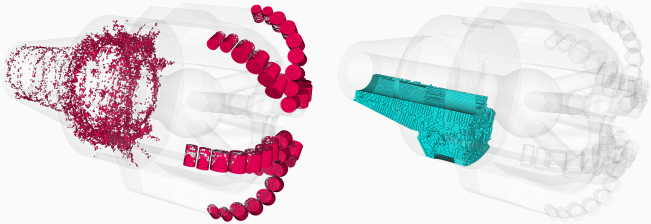
9 0.00 0.00 0.00
10 1.00 0.00 0.00
11 2.00 0.00 0.00

BLACK 0 1 3 6 9 10
LEFT 0 2 4
RIGHT 1 3 5
FACE 6 7 8

Processing of Parsed Data – Clusterization

Hilbert space filling curve:

- A geometric decomposition method (in fact, just another parallel sort)
- Assures spatial locality
- **Minimize the number of neighboring processes**

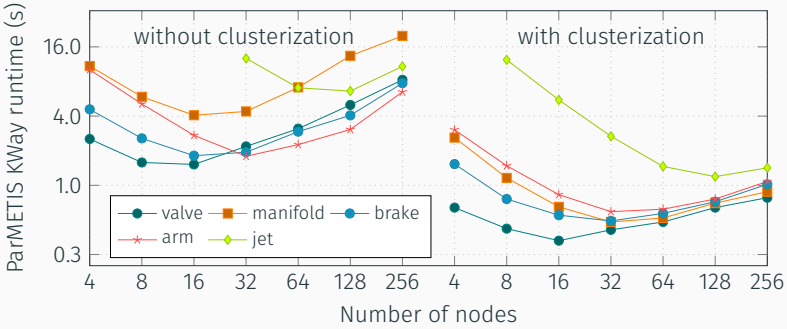


Processing of Parsed Data – the final decomposition

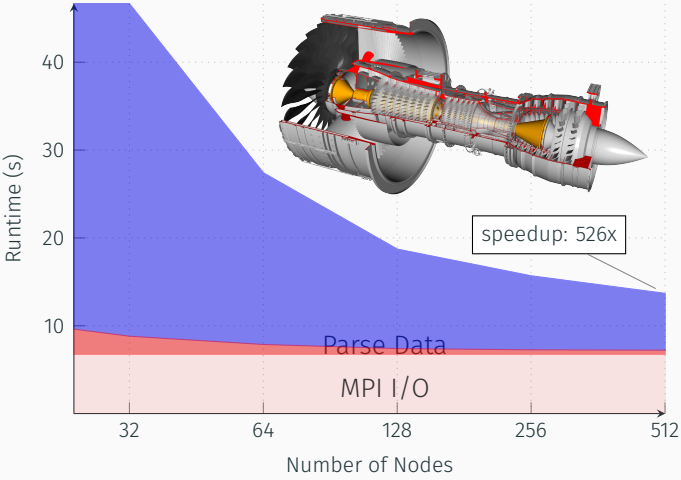
Clusterization significantly improve ParMETIS performance

Best times ratios: valve=4.6, manifold=8.5, brake=3.7, arm=3.1, jet=5.6

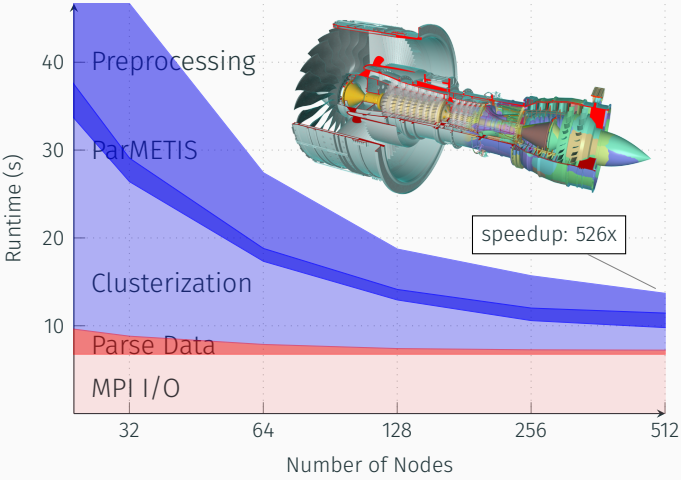
Independent on the mesh geometry



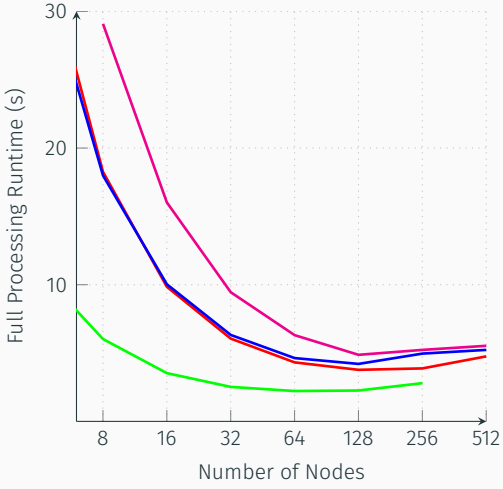
Processing of the Jet Engine example up to 6144 MPI processes



Processing of the Jet Engine example up to 6144 MPI processes

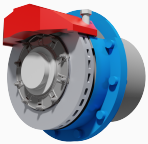


Processing of other examples



Valve (6.3 GB)
433 s → **2.22 s (195x)**

Manifold (21 GB)
1359 s → **3.77 s (360x)**



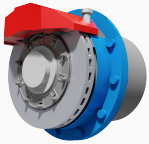
Disc Brake (19 GB)
938 s → **4.21 s (223x)**

Loader Arm (28 GB)
1270 s → **4.87 s (261x)**

A scalable solver does not mean the scalable workflow!

Valve, 6.3 GB / 31M / 23M
433 s → **2.22 s (195x)**

Manifold, 21 GB / 101M / 73M
1359 s → **3.77 s (360x)**



Disc Brake, 19 GB / 122M / 45M
938 s → **4.21 s (223x)**

Loader Arm, 28 GB / 169M / 85M
1270 s → **4.87 s (261x)**

Jet Engine	
file size	142 GB
nodes	822 M
elements	484 M
seq. processing	∼*7200* s
par. processing	**13.69 s**
speedup	**526**