



<i>opt</i>	<i>#cpus</i>	<i>time [s]</i>
01	2048	35.13
01	4096	18.14
01	8192	9.32
03	2048	29.00
03	4096	15.00
03	8192	7.68

JUBE: AN ENVIRONMENT FOR SYSTEMATIC BENCHMARKING AND SCIENTIFIC WORKFLOWS

APR 23RD 2026 | EASYBUILD USER MEETING 2026 @ GUIMARÃES | THOMAS BREUER

MOTIVATION

- HPC is based on workflows and benchmarks
 - Requiring systematic execution of several (dependent) steps
- User groups:
 - system administrators → ensure system functionality and stability; procurements
 - support staff → reproduce problems; detect changes in system behavior
 - HPC users → ensure quality and functionality of their software; scientific use cases
- Technical Implementation:
 - Manual benchmarking 
 - Benchmark-specific script solution 
 - Generic software to configure benchmarks →



WHAT IS JUBE?

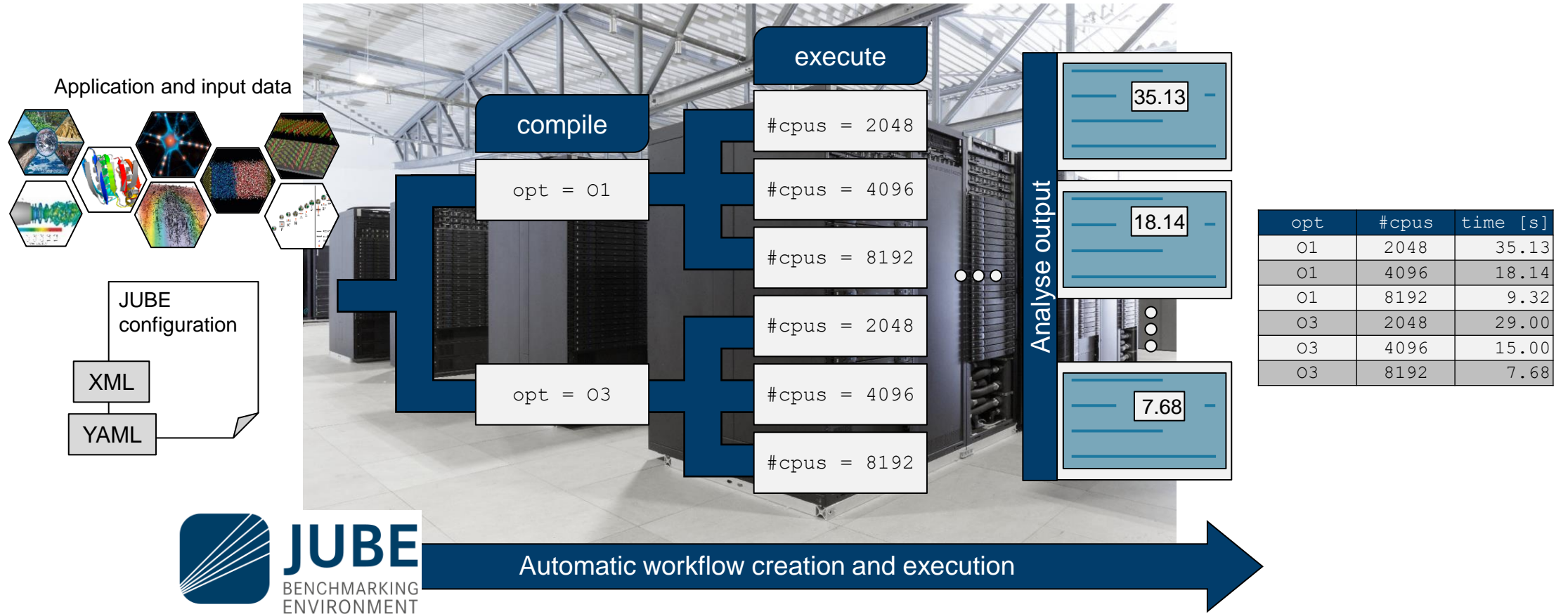
- Generic, lightweight, configurable environment to run, monitor and analyse application execution in a systematic way.
- Can be used for:
 - Workflows, Benchmarking, Parameter studies, Production scenarios, Testing, Profiling, ...
- Open source (GPL-3.0): <https://github.com/FZJ-JSC/JUBE>



jube.fz-juelich.de



GENERIC SCALING WORKFLOW EXAMPLE IN JUBE



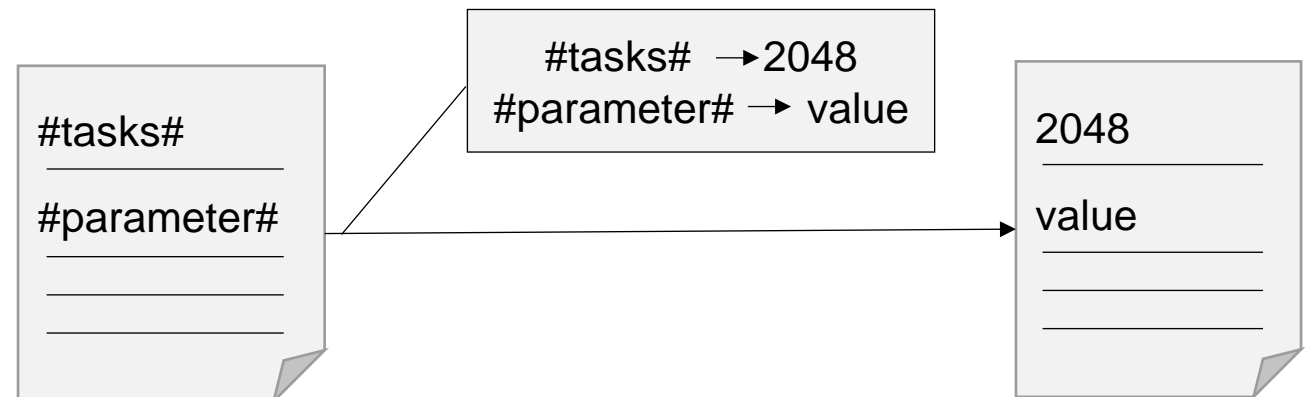
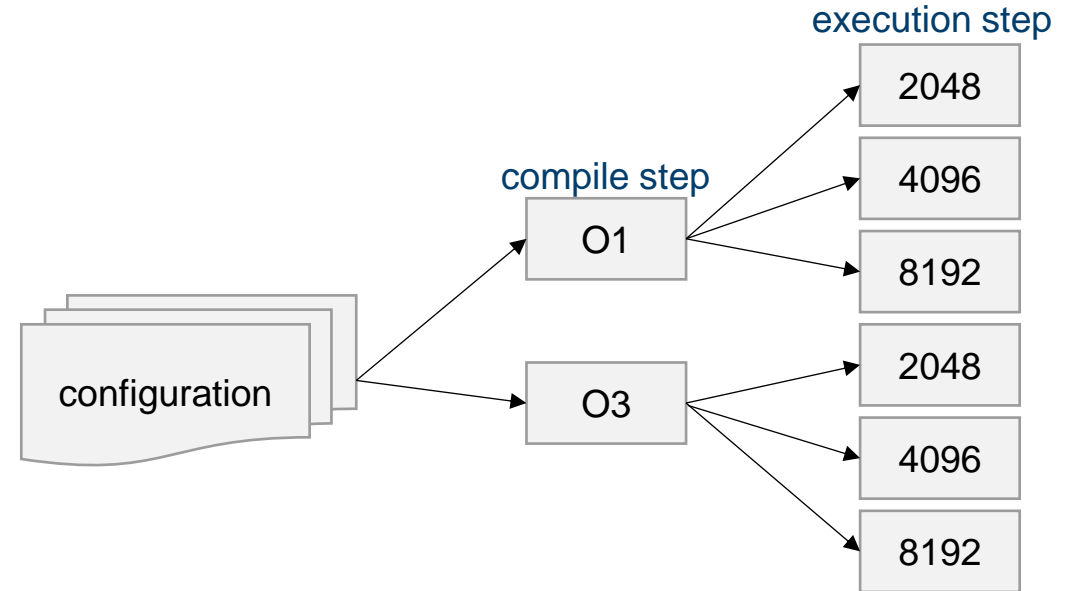
KEY CONCEPTS

- Easy parameter space creation

```
<parameter name="opt">01,03</...>  
<parameter name="tasks">2048,4096,8192</...>
```

- Substitution of placeholders

```
<sub source="#task#" dest="$tasks" />  
<sub source="#parameter#" dest="value" />
```



WORKFLOW (WITHOUT EASYBUILD / EESSI)

```

<jube>
  <benchmark name="bench" outpath="./benchmark runs">
    <parameterset name="compileset">
      <parameter name="execname">my_exe</parameter>
      <parameter name="cppflagslist">
        -O1,-O3
      </parameter>
    </parameterset>

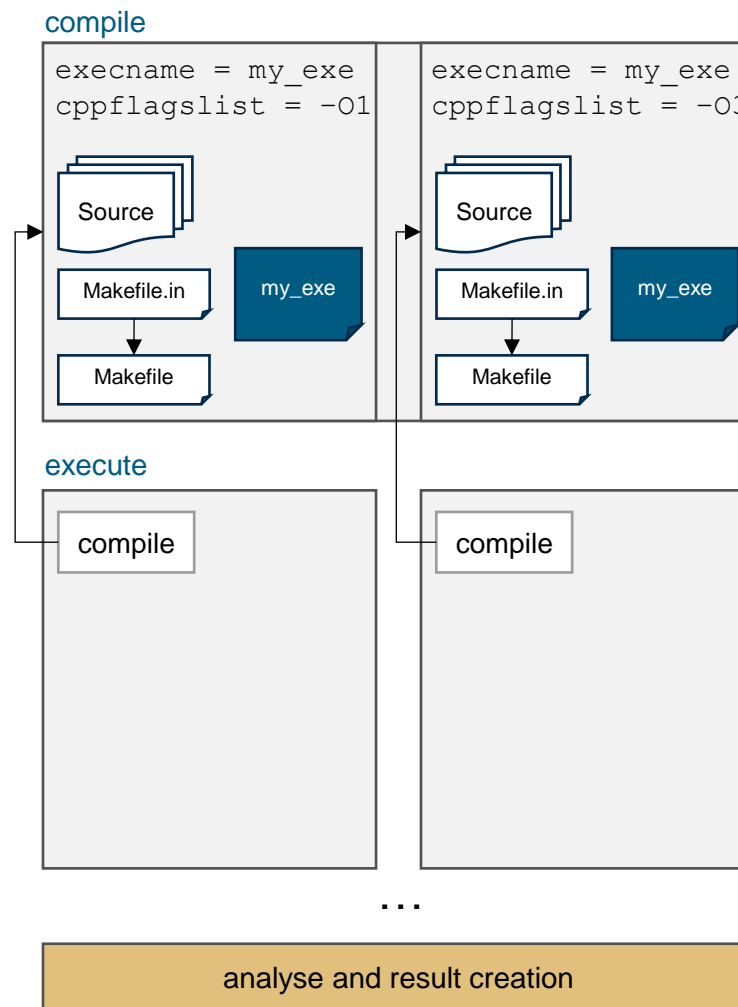
    <fileset name="sources">
      <copy>src/*</copy>
    </fileset>

    <substituteset name="compilesub">
      <iofile in="Makefile.in" out="Makefile" />
      <sub source="#PROGNAME#" dest="$execname" />
    </substituteset>

    <step name="compile">
      <use>compileset</use>
      <use>sources</use>
      <use>compilesub</use>
      <do>make OPT=$cppflagslist</do>
    </step>

    <step name="execute" depend="compile">
      ...
    </step>
  </benchmark>

```



WORKFLOW COMPILE WITH EASYBUILD



```

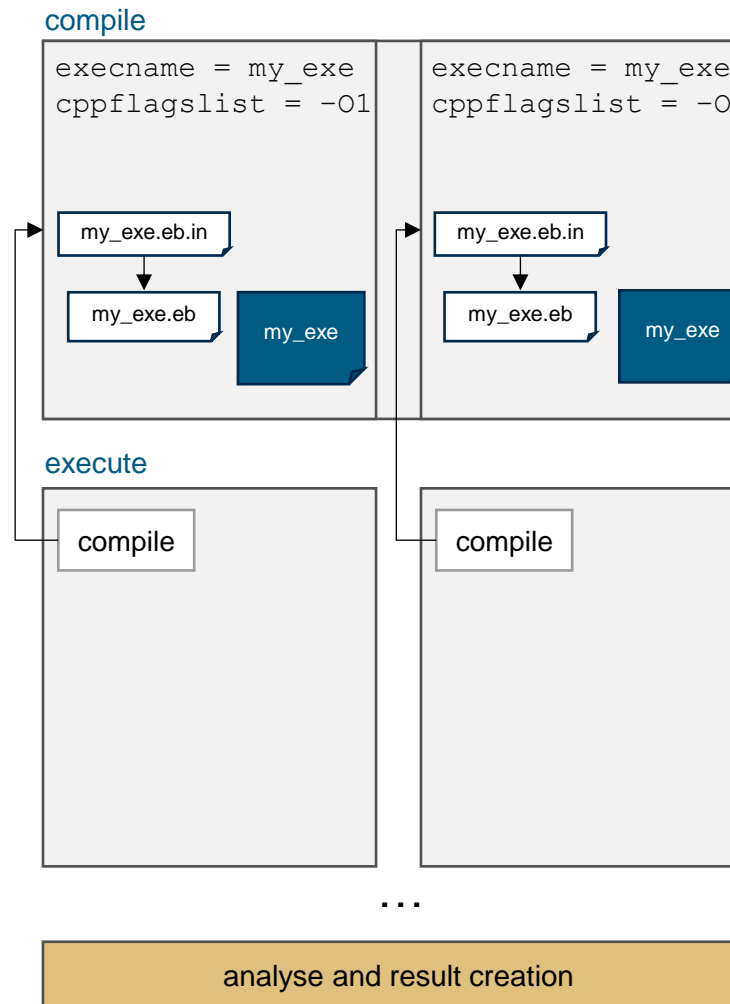
<jube>
  <benchmark name="bench" outpath="./benchmark runs">
    <parameterset name="compileset">
      <parameter name="execname">my_exe</parameter>
      <parameter name="cppflagslist">
        -O1,-O3
      </parameter>
    </parameterset>

    <fileset name="eb_file">
      <copy>${execname}.eb.in</copy>
    </fileset>

    <substituteset name="compilesub">
      <iofile in="${execname}.eb.in" out="${execname}.eb" />
      <sub source="#OPT#" dest="$cppflagslist" />
    </substituteset>

    <step name="compile">
      <use>compileset</use>
      <use>eb_files</use>
      <use>compilesub</use>
      <do>eb ${execname}.eb</do>
    </step>

    <step name="execute" depend="compile">
      ...
  </benchmark>
  
```



WORKFLOW WITH EASYBUILD OR EESSI



```

<jube>
  <benchmark name="bench" outpath="./benchmark runs">
    <parameterset name="executeset">
      <parameter name="execname">my_exe</parameter>
      <parameter name="cppflagslist">
        -O1,-O3
      </parameter>
    </parameterset>

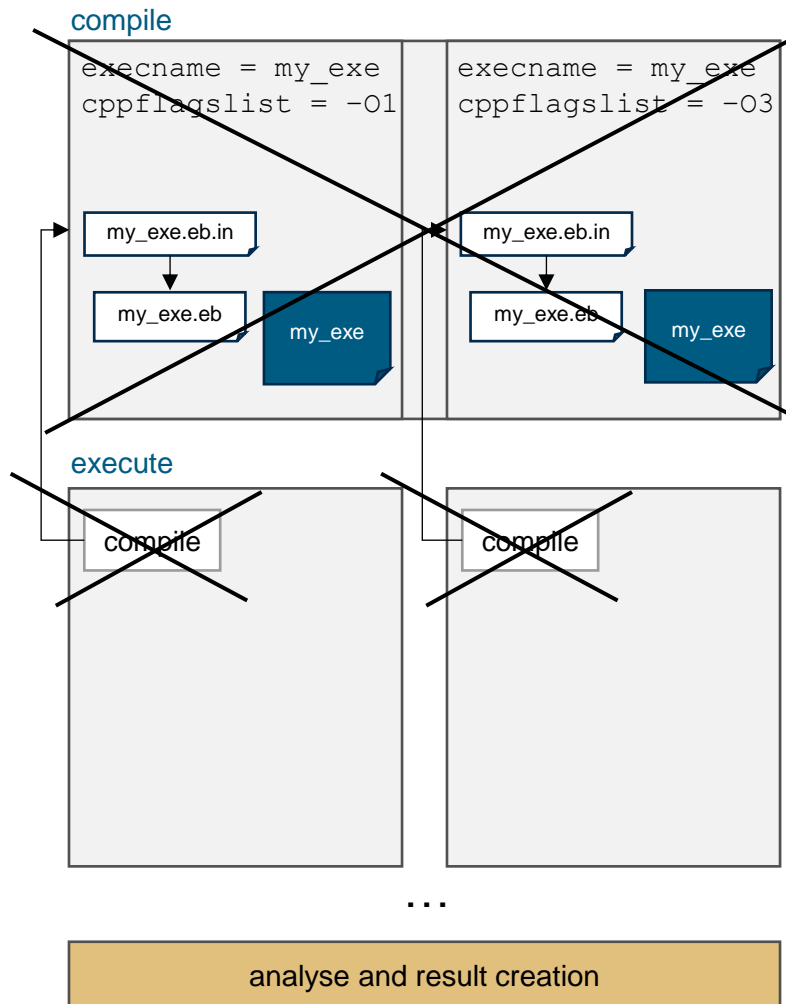
    <fileset name="eb file">
      <copy>${execname}.eb.in</copy>
    </fileset>

    <substituteset name="compilesub">
      <iofile in="${execname}.eb.in" out="${execname}.eb" />
      <sub source="#OPT#" dest="$cppflagslist" />
    </substituteset>

    <step name="compile">
      <use>compileset,eb_files,compilesub</use>
      <do>eb ${execname}.eb</do>
    </step>

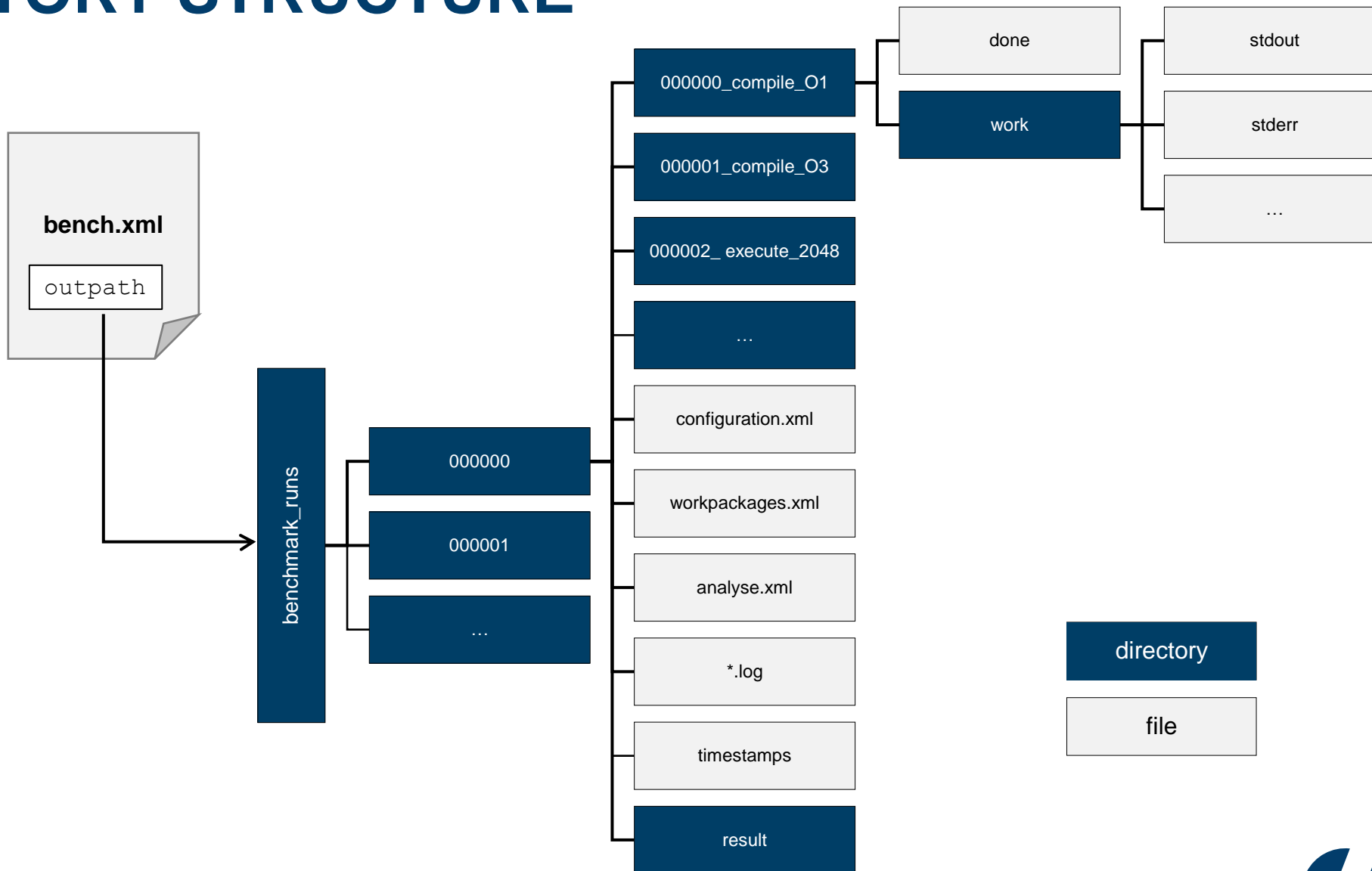
    <step name="execute" depend "compile">
      <use>executeset</use>
      <do>module load ${execname}${cppflagslist}</do>
      ...
    </step>
  </benchmark>

```



DIRECTORY STRUCTURE

example



COMMAND LINE ACCESS



- Start a new benchmark run

```
jube run benchmark.xml [...]
```

- Info mode

```
jube info benchmark_dir [--id <id>] [--step <stepname>]
```

- Command line accessible glossary

```
jube help [<keyword>]
```

- and more ...

```
jube continue, analyse, result, log, ...
```

- for automated workflows via CI

```
jube-autorun [...] benchmark.xml
```



APPLICATIONS OF JUBE

HISTORY OF JUBE IN EUROPEAN BENCHMARKING

- European Benchmarking based on JUBE:
 - 2006-2010: DEISA benchmarking suite
 - 2008-2012: PABS: PRACE application benchmark suite (PRACE-PP/1IP)
 - >2012: PUABS: PRACE Unified Application benchmark suite (PRACE-2IP)
- Examples of inter-/national projects using JUBE:



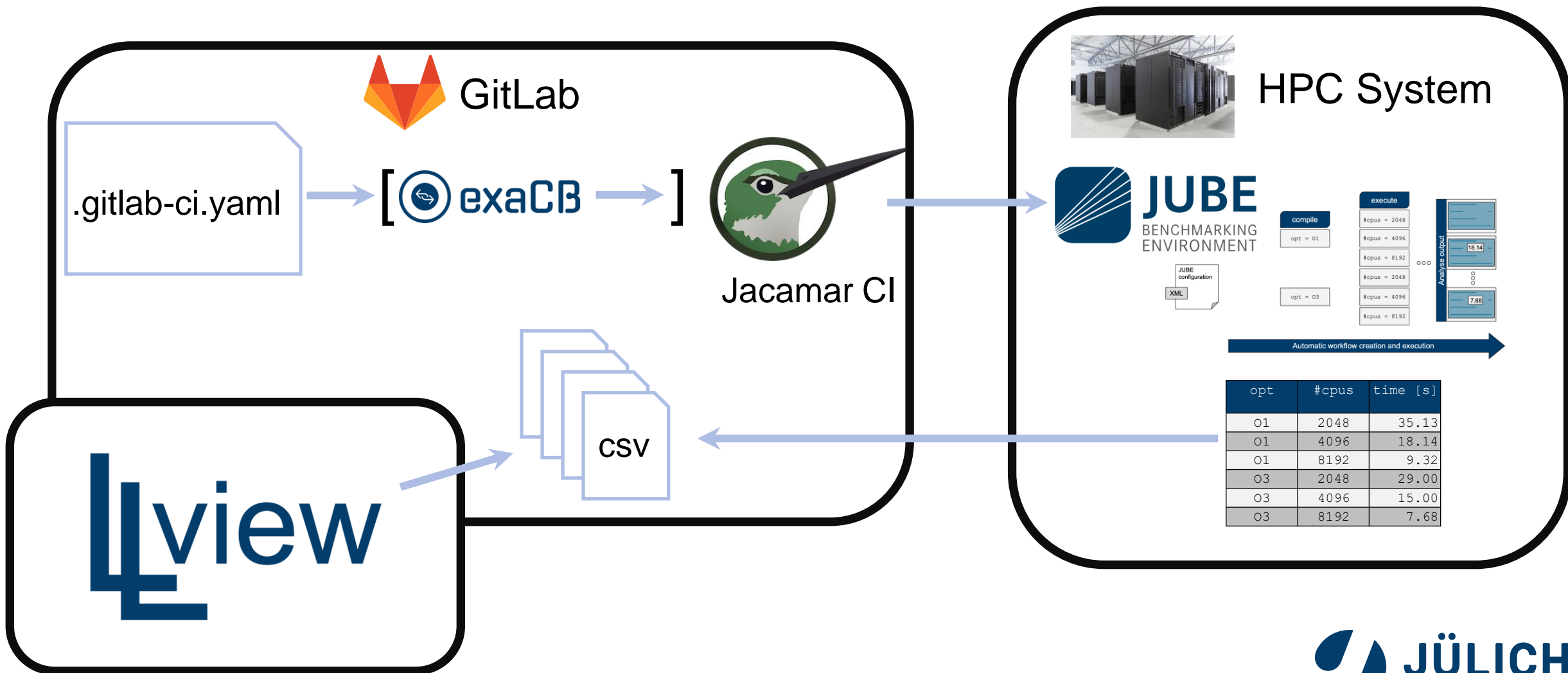
EXAMPLE USE CASE: JUPITER BENCHMARK SUITE



- Standardized JUBE scripts for 16 Application & 8 synthetic benchmarks
- Execution targets:
 - JUPITER Booster (GPU, CPU)
 - JUPITER Cluster (CPU)
- Application benchmark categories:
 - Total Cost of Ownership (TCO)
 - High-Scaling
- Published on GitHub:
 - <https://github.com/FZJ-JSC/jubench>

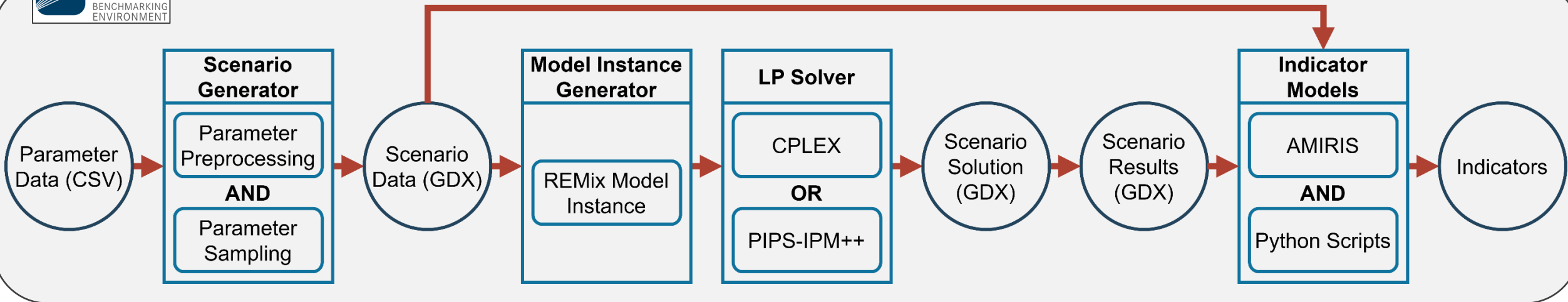
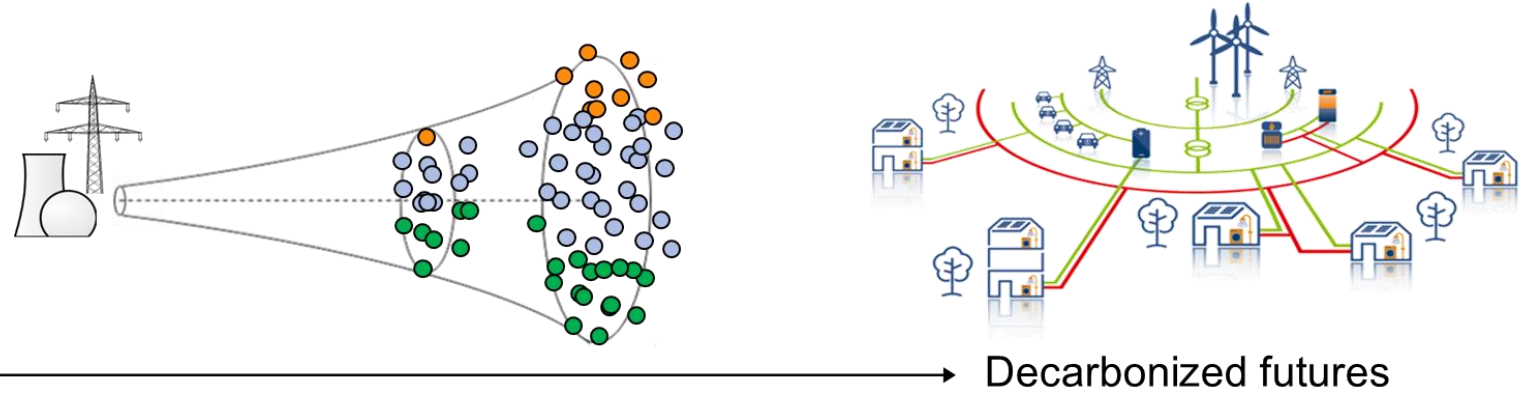
Name	Booster			Cluster
	GPU	GPU High-Scale	CPU	CPU
Amber	✓			
Arbor	✓	✓		
Chroma	✓	✓		
Gromacs	✓			
ICON	✓			
JUQCS	✓	✓		
nekRS	✓	✓		
ParFlow	✓			
PConGPU	✓	✓		
Quantum ESPRESSO	✓			
SOMA	✓			
AI-MMoCLIP	✓			
AI-NLP	✓			
AI-ResNet	✓			
dynQCD				✓
NASTJA				✓
Graph500			✓	
HPCG	✓			✓
HPL	✓			✓
IOR			✓	✓
LinkTest			✓	✓
Multi-Flow IP			✓	
OSU	✓		✓	✓
STREAM	✓			✓

EXAMPLE USE CASE: CONTINUOUS BENCHMARKING FOR HPC SYSTEM HEALTH



EXAMPLE USE CASE: ENERGY SYSTEM MODELLING

Evaluation of more than 11,000 scenarios for the German power system



CURRENT STATUS

- Main development in internal GitLab
 - Many open issues (almost all are feature requests)
- Cannot be published on pypi as “jube“ (prohibited name 😞) [1]

OUTLOOK

- Release version 3: status database, figure, GUI, ...
- Extend & improve documentation and testing, implement new features
- Continue the transition to GitHub

JUBE AVAILABILITY

- Online documentation, (adv.) tutorial, FAQ, glossary:

go.fzj.de/jube

- Carpentries:

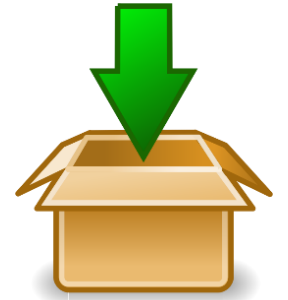
<https://carpentries-incubator.github.io/hpc-workflows-jube>

- Prerequisites:

- OS: Linux
- Python 3

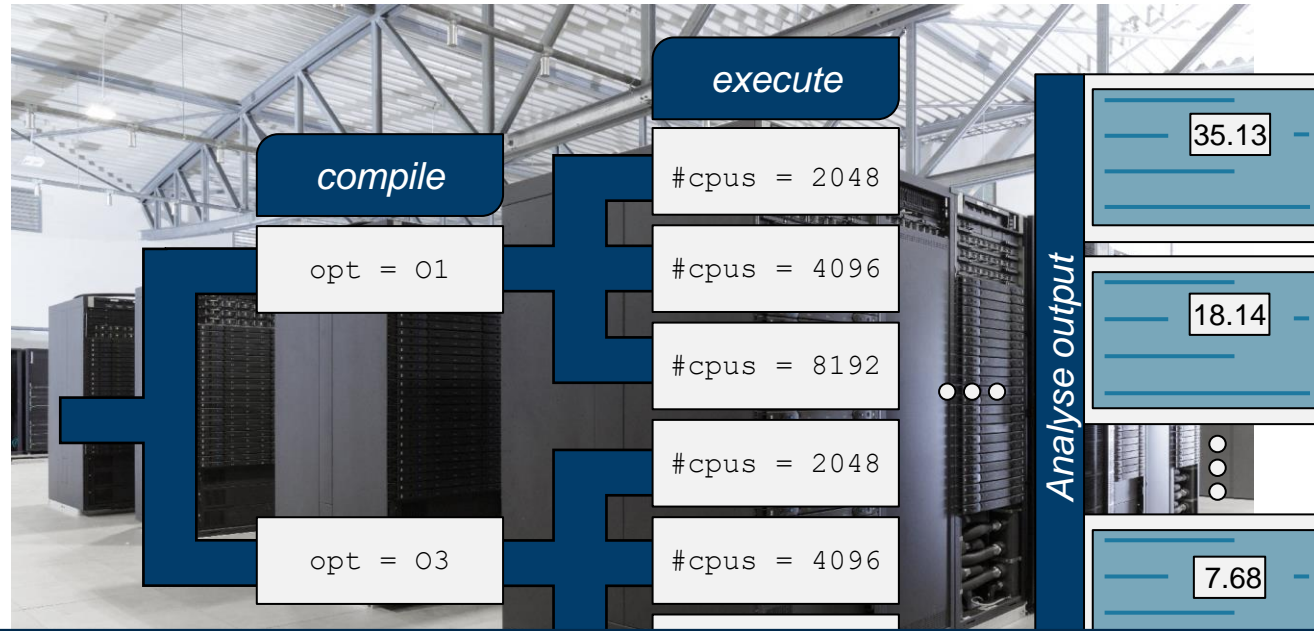
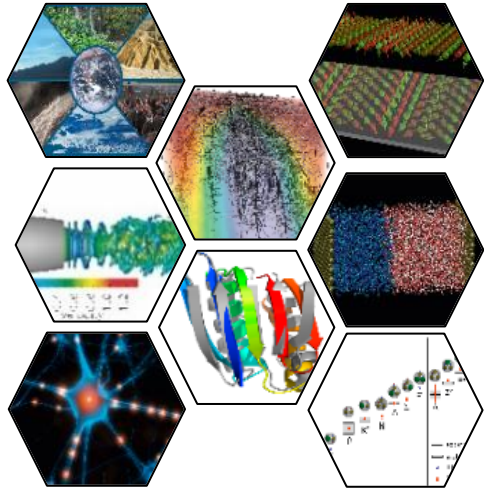
- Contact (developers @JSC):

jube.jsc@fz-juelich.de



EVENTS @ ISC26

- Booth (Foyer 3 + H): Jülich Supercomputing Centre
 - LLview Demo: Job & System Monitoring + Benchmarking Dashboard
- Research Poster and Talk @ MODA26 Workshop:
 - Centralised Dashboard for Continuous Benchmarking: From HPC Clusters to Quantum Processors
- BOF: Thu 25th June - 2:15 PM – Hall F (2nd floor)
 - Rethinking HPC User Support for the AI and Exascale Era
- ...



<i>opt</i>	<i>#cpus</i>	<i>time [s]</i>
01	2048	35.13
01	4096	18.14
01	8192	9.32
03	2048	29.00
03	4096	15.00
03	8192	7.68

THANK YOU!

JUBE



jube.jsc@fz-juelich.de

Thomas Breuer
LinkedIn:



t.breuer@fz-juelich.de