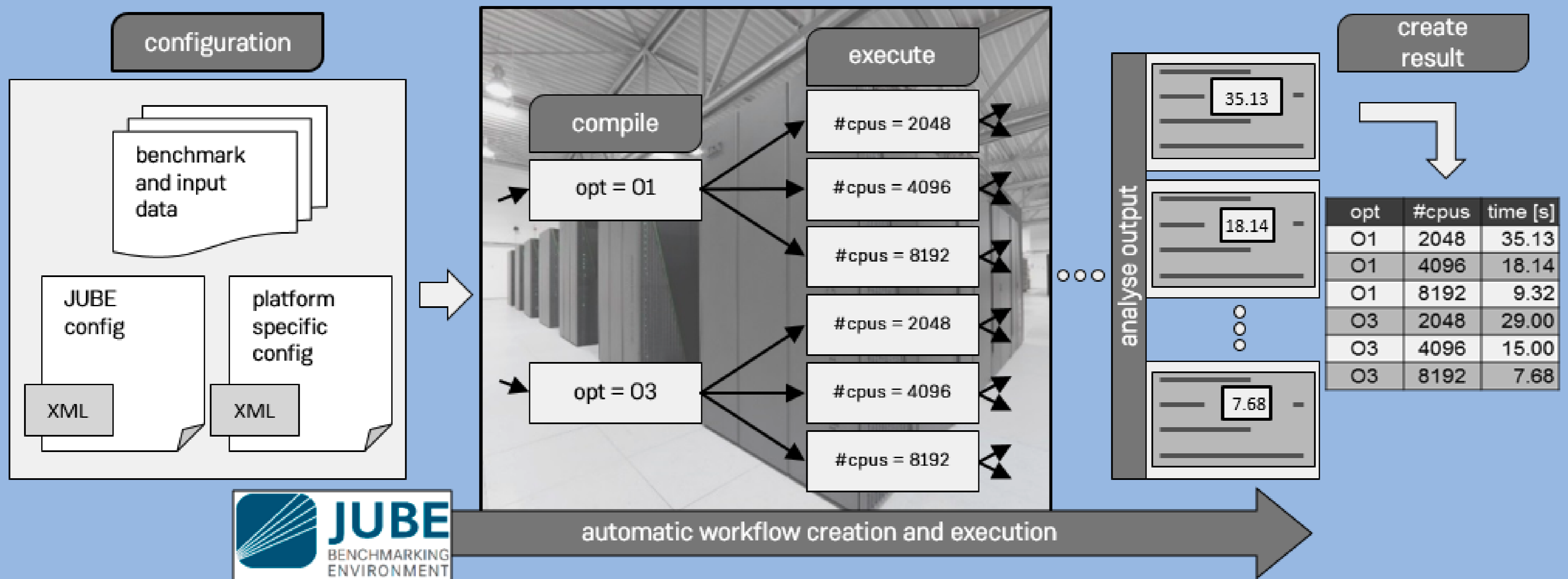


# JUBE

## A FLEXIBLE, APPLICATION- AND PLATFORM-INDEPENDENT ENVIRONMENT FOR BENCHMARKING



JUBE considerably **reduces the cost** of configuring, running and analysing benchmark, test or production suites

### Flexible and generic workflow management

XML- or YAML-file based configuration allows a very **flexible parameterization**

Use cases: Procurements, comparison of systems, monitoring effects of system and configuration changes, software testing

### Key concepts and strategy

- Provision of templates: Independence of applications and platforms
- Span a multi-dimensional parameter space
- Substitution of placeholders
- Workflow and directory management
- No source code modifications and configuration files created once
- Reproducibility

```
<jube>
<benchmark name="bench" outpath="./runs">
  <parameterset name="compileset">
    <parameter name="execn">my_exe</parameter>
    <parameter name="cppflag">
      -O1, -O2
    </parameter>
  </parameterset>
  <fileset name="sources">
    <copy>src/*</copy>
  </fileset>
  <substituteset name="compilesub">
    <iofile in="Makefile.in" out="Makefile" />
    <sub source="#PROGNAME#" dest="$execn" />
  </substituteset>
  <step name="compile">
    <use>compileset, sources, compilesub</use>
    <do>make OPT=$cppflag</do>
  </step>
  <step name="execute" depend="compile">...
```

```
> jube run config.xml -r
# benchmark: bench
# id: 0
| stepname | all | open | wait | error | done |
|-----|-----|-----|-----|-----|-----|
| compile  | 2   | 0   | 0   | 0   | 2   |
| execute  | 6   | 0   | 0   | 0   | 6   |

result:
| cppflag | cpus | time[s] |
|-----|-----|-----|
| -O1     | 2048 | 35.13   |
| -O1     | 4096 | 18.14   |
...
```

### Implementation and Availability

- Implemented in Python
- Open Source (GPLv3)
- Documentation and tutorial available online

