

AiiDA SIRIUS

Managing high-throughput computational science

Towards exascale with AiiDA and SIRIUS-Quantum ESPRESSO

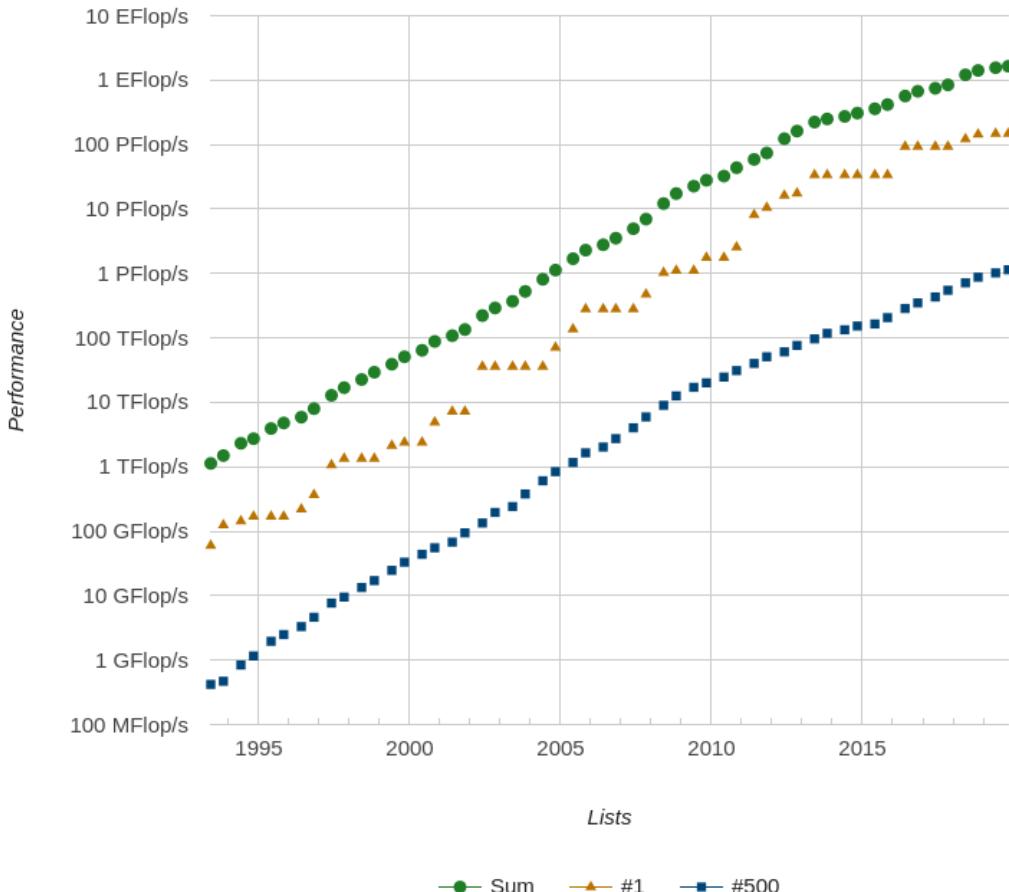
Sebastiaan Huber, École Polytechnique Fédérale de Lausanne
MaX webinar, 27th of May 2020



"exascale", has received funding from the European Union's Horizon 2020 project call H2020-INFRAEDI-



DEVELOPMENT OF COMPUTATIONAL POWER



- > Steady increase of available computational power over last 25 years [1]
- > Calculation that took months in 1995, takes seconds in 2020
- > Calculations on a single material in 1995, now we run on tens of thousands

[1] <https://www.top500.org/>

Workflow automation

- Need tools to define complex workflows with advanced error handling
- An automated, robust and scalable engine to run the workflows

Data management

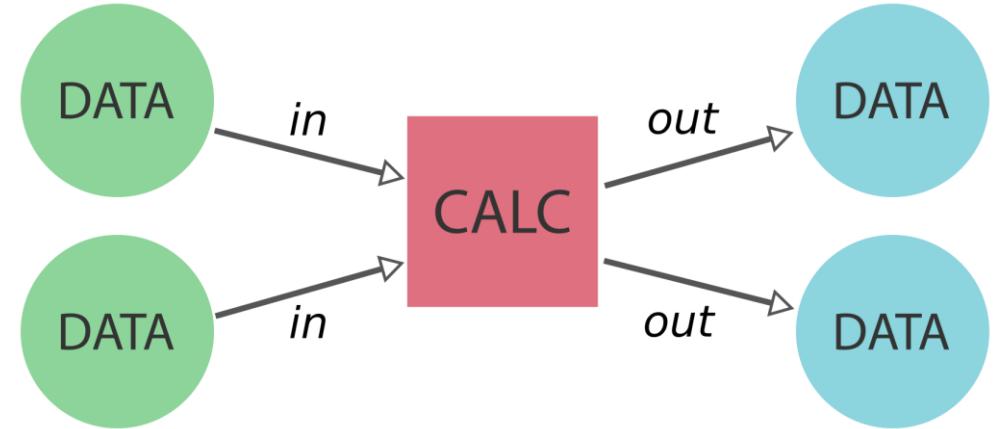
- Data should be stored reliably and efficiently
- Stored data should be interoperable and queryable

Reproducibility

- All produced data should be reproducible by storing the full provenance

Simple recipe

- Store data transformations or ‘calculations’
- Store its **inputs** and their metadata
- Store its **outputs** and their metadata
- Most crucially store the **inter-connections**

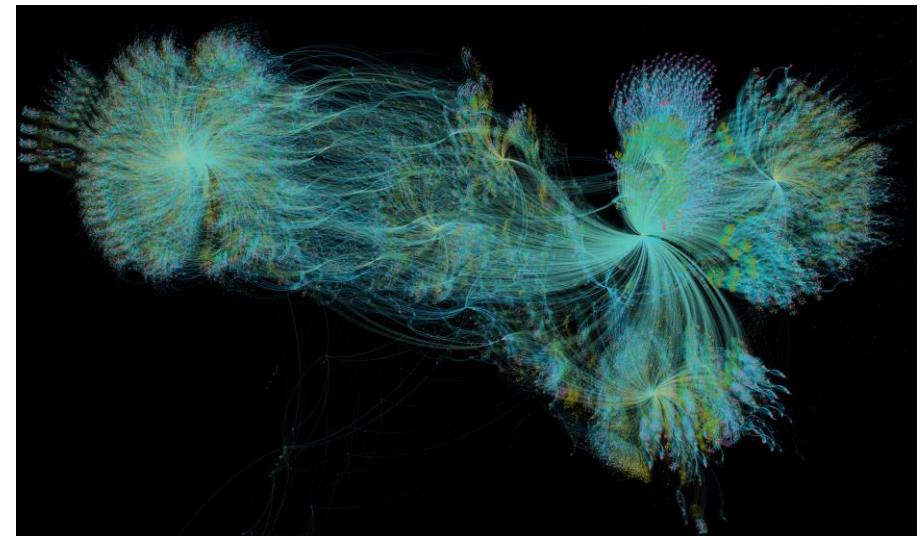
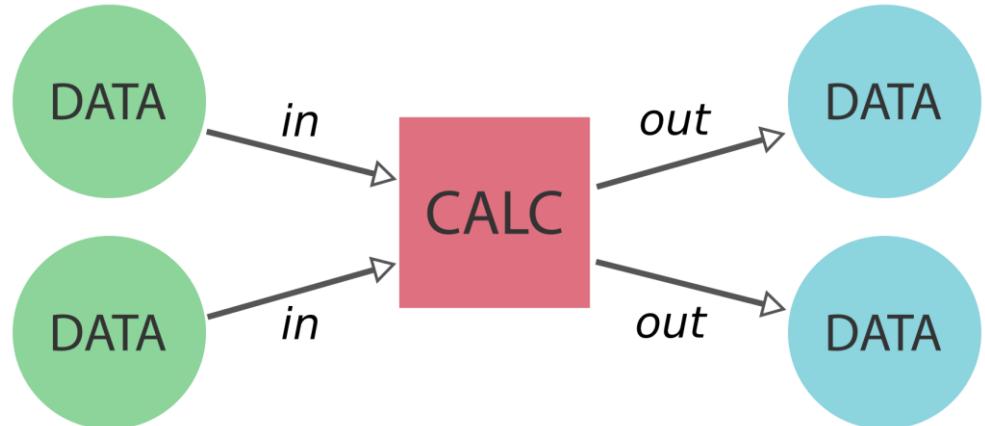


Simple recipe

- Store data transformations or ‘calculations’
- Store its **inputs** and their metadata
- Store its **outputs** and their metadata
- Most crucially store the **inter-connections**

Requirements

- Needs to be automated
- Needs to be stored *as data is created*



Graphical representation of actual AiiDA database
courtesy of Jens Bröder, Forschungszentrum Jülich, Germany

DATA PROVENANCE

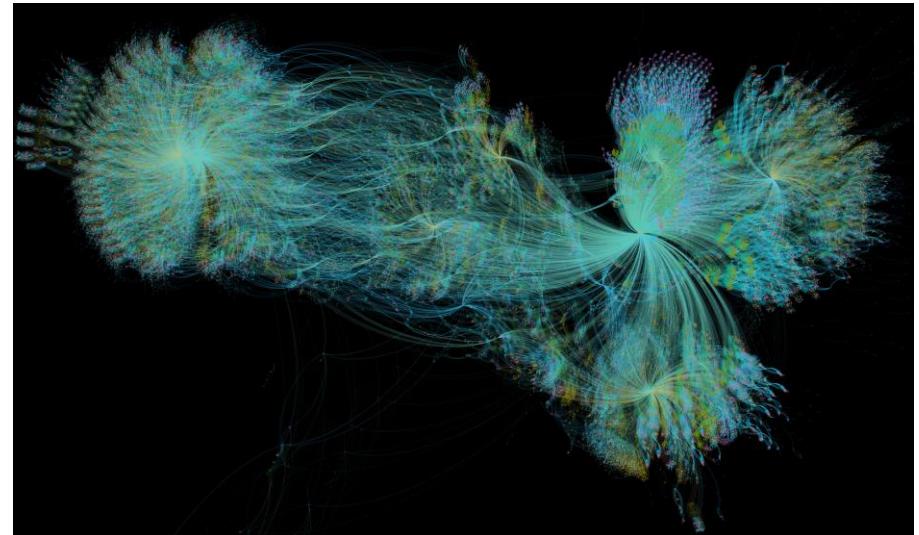
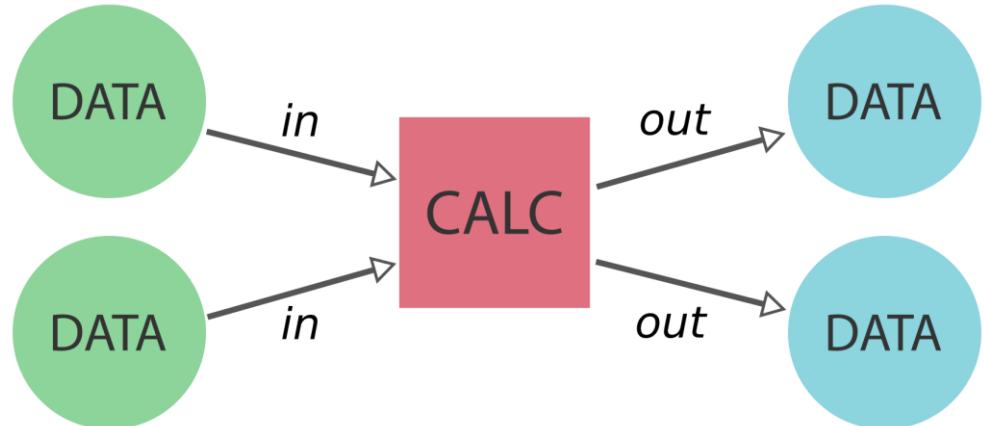
Simple recipe

- Store data transformations or ‘calculations’
- Store its **inputs** and their metadata
- Store its **outputs** and their metadata
- Most crucially store the **inter-connections**

Requirements

- Needs to be automated
- Needs to be stored *as data is created*

Complexity grows quickly even for simple workflows and is impossible to reconstruct
a posteriori



Graphical representation of actual AiiDA database
courtesy of Jens Bröder, Forschungszentrum Jülich, Germany

AUTOMATED INTERACTIVE INFRASTRUCTURE AND DATABASE



DRIVING REPRODUCIBLE COMPUTATIONAL SCIENCE TOWARDS EXASCALE



Scalable workflow engine



Automated full data provenance



Built-in support for HPC



Flexible plugin system



Language: implemented and API in python



License: MIT open source <http://www.aiida.net>



Source: <https://github.com/aiidateam/aiida-core>

RUNNING QE WITH AIIDA

- Python API to submit calculations
- Run automatically by the daemon

```
from aiida import orm
inputs = {
    'code': orm.load_code('qe@localhost'),
    'structure': orm.StructureData.from_cif('Si.cif'),
    'kpoints': orm.KpointsData([2, 2, 2]),
    'parameters': orm.Dict(dict={
        'CONTROL': {'calculation': 'scf'},
        'SYSTEM': {'ecutwfc': 40, 'ecutrho': 320},
    }),
    'metadata': {
        'options': {
            'resources': {
                'num_machines': 1
            },
            'max_wallclock_seconds': 1800
        }
    }
}
submit(PwCalculation, **inputs)
```

RUNNING QE WITH AIIDA

- Python API to submit calculations
- Run automatically by the daemon

```
from aiida import orm
inputs = {
    'code': orm.load_code('qe@localhost'),
    'structure': orm.StructureData.from_cif('Si.cif'),
    'kpoints': orm.KpointsData([2, 2, 2]),
    'parameters': orm.Dict(dict={
        'CONTROL': {'calculation': 'scf'},
        'SYSTEM': {'ecutwfc': 40, 'ecutrfho': 320},
    }),
    'metadata': {
        'options': {
            'resources': {
                'num_machines': 1
            },
            'max_wallclock_seconds': 1800
        }
    }
}
submit(PwCalculation, **inputs)
```

- Submitted calculations automatically stored in the database

PK	Created	Process label	Process State	Process status
10170	1m ago	PwCalculation	■ Finished [0]	

Total results: 1

Info: last time an entry changed state: 1m ago (at 18:37:54 on 2020-05-24)

RUNNING QE WITH AIIDA

- Python API to submit calculations
- Run automatically by the daemon

```
from aiida import orm
```

```
inputs = {  
    'code': orm.load_code('qe@localhost'),  
    'structure': orm.StructureData.from_cif('Si.cif'),  
    'kpoints': orm.KpointsData([2, 2, 2]),  
    'parameters': orm.Dict(dict={  
        'CONTROL': {'calculation': 'scf'},  
        'SYSTEM': {'ecutwfc': 40, 'ecutrfho': 320},  
    }),  
    'metadata': {  
        'options': {  
            'resources': {  
                'num_machines': 1  
            },  
            'max_wallclock_seconds': 1800  
        }  
    }  
}  
  
submit(PwCalculation, **inputs)
```

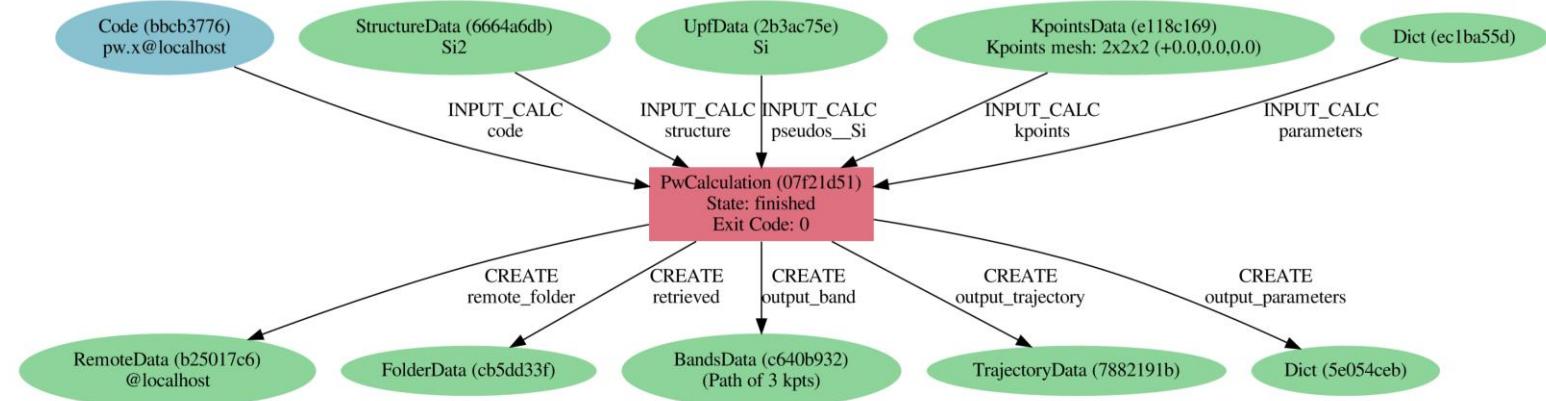
- Submitted calculations automatically stored in the database

PK	Created	Process label	Process State	Process status
10170	1m ago	PwCalculation	Finished [0]	

Total results: 1

Info: last time an entry changed state: 1m ago (at 18:37:54 on 2020-05-24)

- Full provenance: inputs, outputs and raw input/output data are stored



RUNNING QE WITH AIIDA

- Python API to submit calculations
- Run automatically by the daemon

```
from aiida import orm
```

```
inputs = {  
    'code': orm.load_code('qe@piz-daint')  
    'structure': orm.StructureData.from_cif('Si.cif'),  
    'kpoints': orm.KpointsData([2, 2, 2]),  
    'parameters': orm.Dict(dict={  
        'CONTROL': {'calculation': 'scf'},  
        'SYSTEM': {'ecutwfc': 40, 'ecutrfho': 320},  
    }),  
    'metadata': {  
        'options': {  
            'resources': {  
                'num_machines': 1  
            },  
            'max_wallclock_seconds': 1800  
        }  
    }  
}  
  
submit(PwCalculation, **inputs)
```

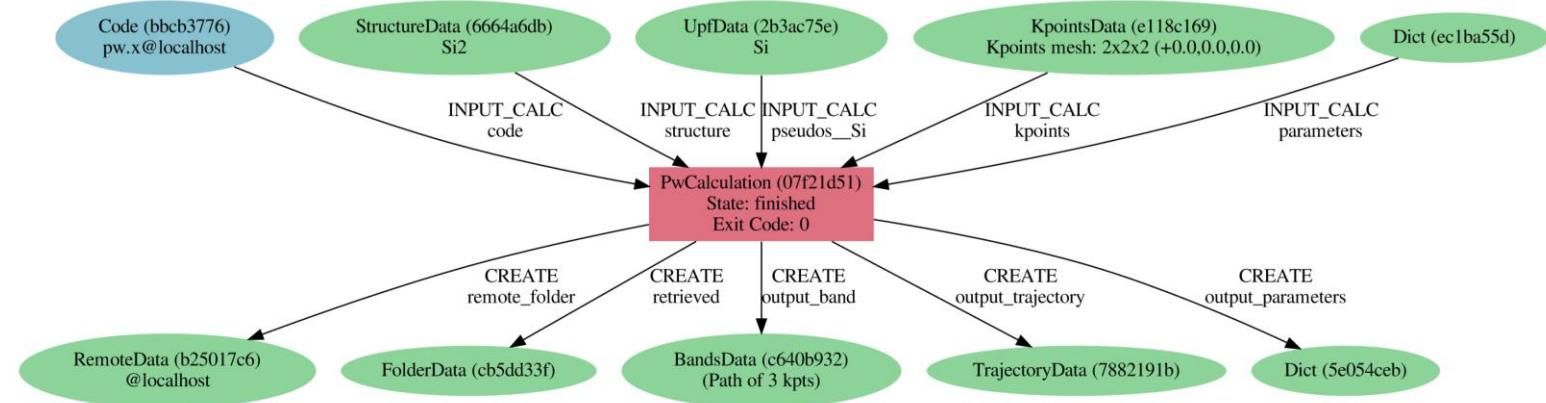
- Submitted calculations automatically stored in the database

PK	Created	Process label	Process State	Process status
10170	1m ago	PwCalculation	■ Finished [0]	

Total results: 1

Info: last time an entry changed state: 1m ago (at 18:37:54 on 2020-05-24)

- Full provenance: inputs, outputs and raw input/output data are stored



- Submit to a different computer by changing a single line

RUNNING QE WITH AIIDA: WORKFLOWS

- AiiDA provides powerful workflow language
- Actual workflows implemented by plugins

```
from aiida import orm
inputs = {
    'pw': {
        'code': orm.load_code('qe@piz-daint'),
        'structure': orm.StructureData.from_cif('Si.cif'),
        'parameters': orm.Dict(dict={
            'CONTROL': {'calculation': 'scf'},
            'SYSTEM': {'ecutwfc': 40, 'ecutrfho': 320},
        }),
        'metadata': {
            'options': {
                'max_wallclock_seconds': 1800
            }
        }
    },
    'kpoints_distance': orm.Float(0.2)
}
submit(PwBaseWorkChain, **inputs)
```

RUNNING QE WITH AIIDA: WORKFLOWS

- AiiDA provides powerful workflow language
- Actual workflows implemented by plugins

```
from aiida import orm
inputs = {
    'pw': {
        'code': orm.load_code('qe@piz-daint')
        'structure': orm.StructureData.from_cif('Si.cif'),
        'parameters': orm.Dict(dict={
            'CONTROL': {'calculation': 'scf'},
            'SYSTEM': {'ecutwfc': 40, 'ecutrfho': 320},
        }),
        'metadata': {
            'options': {
                'max_wallclock_seconds': 1800
            }
        }
    },
    'kpoints_distance': orm.Float(0.2)
}
submit(PwBaseWorkChain, **inputs)
```

- Workchains can orchestrate complex workflows with error-handling

PK	Created	Process label	Process State	Process status
10170	2m ago	PwCalculation	■ Finished	[0]
10180	44s ago	PwBaseWorkChain	■ Finished	[0]
10182	43s ago	create_kpoints_from_distance	■ Finished	[0]
10186	41s ago	PwCalculation	■ Finished	[0]
10194	25s ago	PwCalculation	■ Finished	[0]
Total results: 5				
Info: last time an entry changed state: 5s ago (at 18:40:11 on 2020-05-24)				

RUNNING QE WITH AIIDA: WORKFLOWS

- AiiDA provides powerful workflow language
- Actual workflows implemented by plugins

```
from aiida import orm
```

```
inputs = {
    'pw': {
        'code': orm.load_code('qe@piz-daint')
        'structure': orm.StructureData.from_cif('Si.cif'),
        'parameters': orm.Dict(dict={
            'CONTROL': {'calculation': 'scf'},
            'SYSTEM': {'ecutwfc': 40, 'ecutrfho': 320},
        }),
        'metadata': {
            'options': {
                'max_wallclock_seconds': 1800
            }
        }
    },
    'kpoints_distance': orm.Float(0.2)
}
submit(PwBaseWorkChain, **inputs)
```

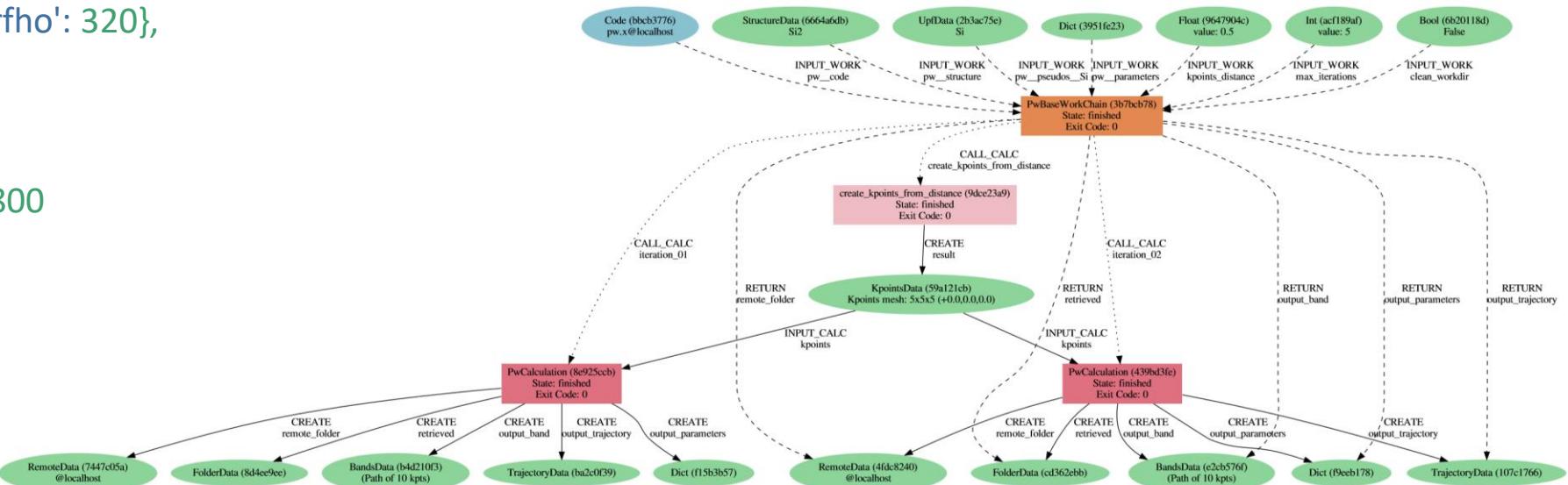
- Workchains can orchestrate complex workflows with error-handling

PK	Created	Process label	Process State	Process status
10170	2m ago	PwCalculation	■ Finished	[0]
10180	44s ago	PwBaseWorkChain	■ Finished	[0]
10182	43s ago	create_kpoints_from_distance	■ Finished	[0]
10186	41s ago	PwCalculation	■ Finished	[0]
10194	25s ago	PwCalculation	■ Finished	[0]

Total results: 5

Info: last time an entry changed state: 5s ago (at 18:40:11 on 2020-05-24)

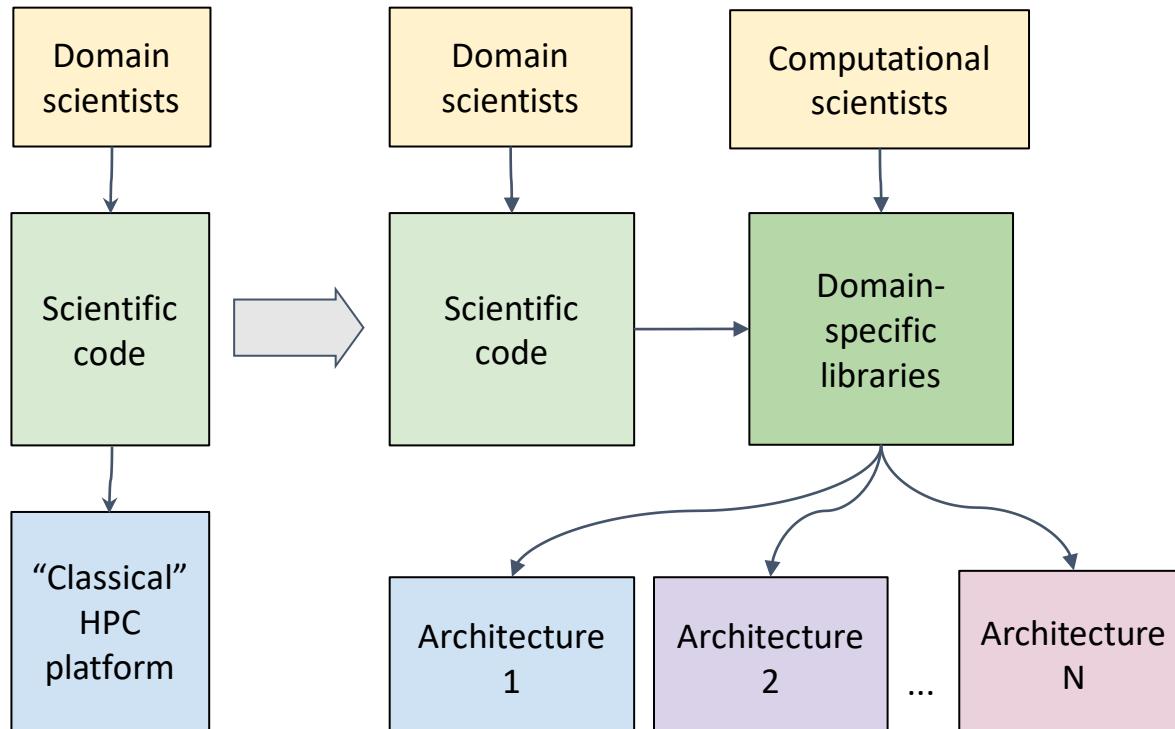
- No matter the complexity: full provenance is automatically stored



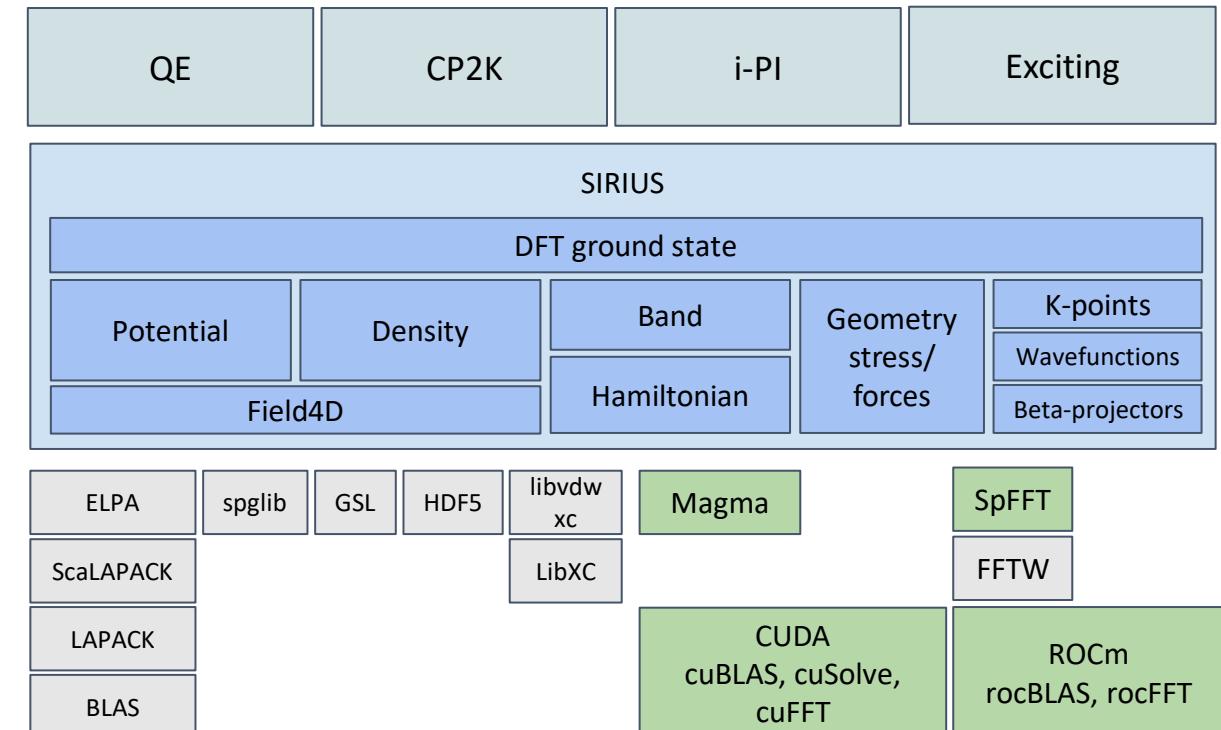
SIRIUS: PLANE-WAVE ELECTRONIC STRUCTURE ON THE GPU

SIRIUS

Library for plane-wave electronic structure codes [1]



Anton Kozhevnikov (CSCS)
Mathieu Taillefumier (CSCS)
Simon Pintarelli (CSCS)
Simon Frasch (CSCS)



QE-SIRIUS: VALIDATION AND BENCHMARKING

- Set of **550 structures** with wide range of chemical and geometrical properties
- Use **aiida-quantumespresso** to run **PwBaseWorkChain** for all structures for each code
- Use **Groups** to store the submitted workchains for easy referencing

Info: to show groups of all types, use the ` -a/-all` option.				
PK	Label	Type string	User	Node count
3	structures/benchmark	core	sebastiaan.huber@epfl.ch	550
4	run/daint-gpu/pw-v6.4	core	sebastiaan.huber@epfl.ch	550
5	run/daint-gpu/pw-v6.4-sirius-v6.5.2	core	sebastiaan.huber@epfl.ch	550

QE-SIRIUS: VALIDATION AND BENCHMARKING

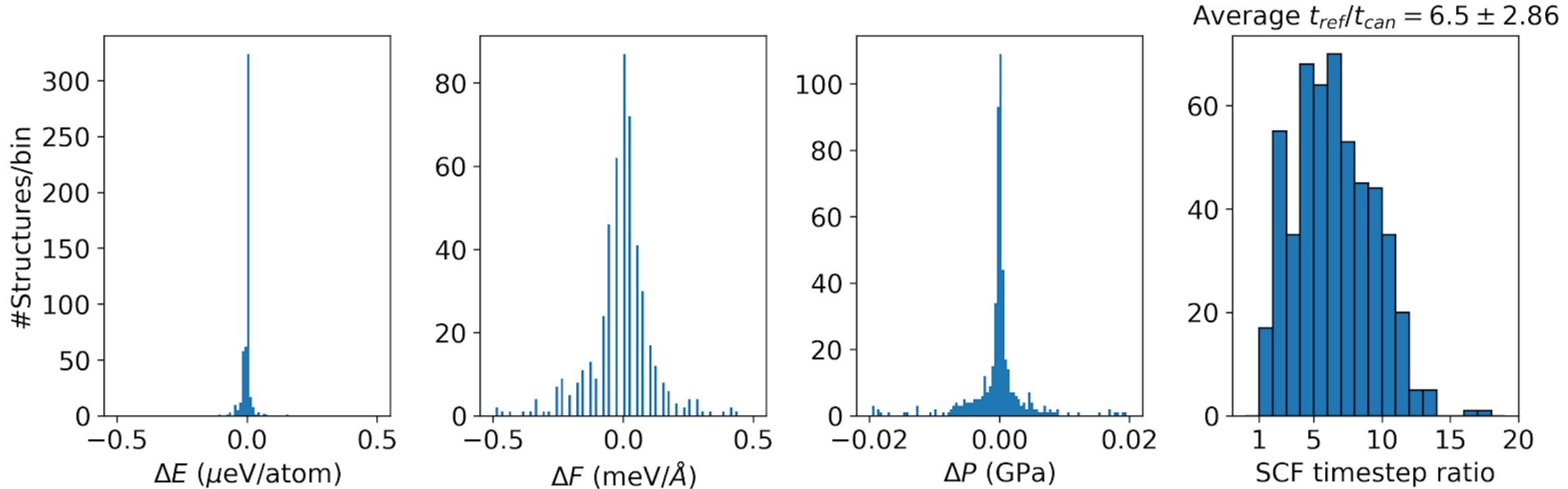
- Set of **550 structures** with wide range of chemical and geometrical properties
- Use **aiida-quantumespresso** to run **PwBaseWorkChain** for all structures for each code
- Use **Groups** to store the submitted workchains for easy referencing

Info: to show groups of all types, use the ` -a/-all` option.					
PK	Label	Type string	User	Node count	
3	structures/benchmark	core	sebastiaan.huber@epfl.ch	550	
4	run/daint-gpu/pw-v6.4	core	sebastiaan.huber@epfl.ch	550	
5	run/daint-gpu/pw-v6.4-sirius-v6.5.2	core	sebastiaan.huber@epfl.ch	550	

- Results of completed workchains can be easily retrieved with the **QueryBuilder**
- Example to **retrieve** all computed **total energies** for **pw-v6.4**

```
builder = QueryBuilder().append(  
    Group, filters={'label': 'run/daint-gpu/pw-v6.4'}, tag='group').append(  
    PwCalculation, with_group='group', tag='pw').append(  
        Dict, with_incoming='pw', project='attributes.total_energy')  
energies = builder.all(flat=True)
```

QE-SIRIUS: VALIDATION AND BENCHMARKING

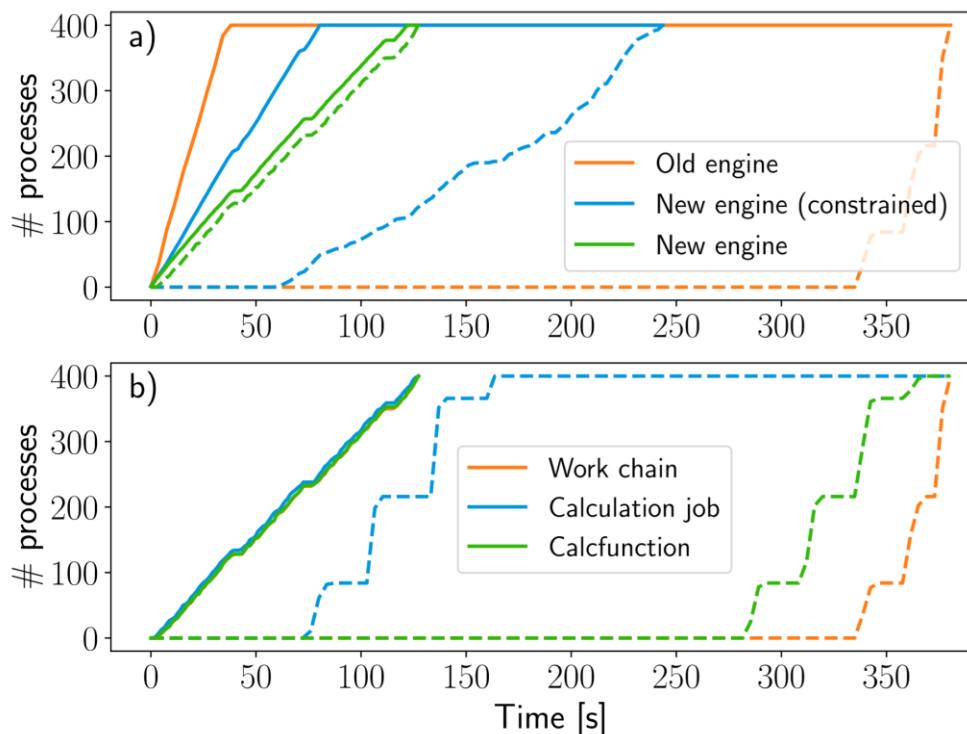


- **Negligible differences** in energies, forces and stress
- Remaining discrepancies due to use of different numerical algorithms
- An average **speedup factor of 6** of QE-SIRIUS with respect to normal QE

HIGH-THROUGHPUT TOWARDS THE EXASCALE

AiiDA 1.0: significant improvements to enable high-throughput computational loads [1]

- Responsive: changing from poll-based to event-based design
- Scalable: arbitrary number of independent daemon workers

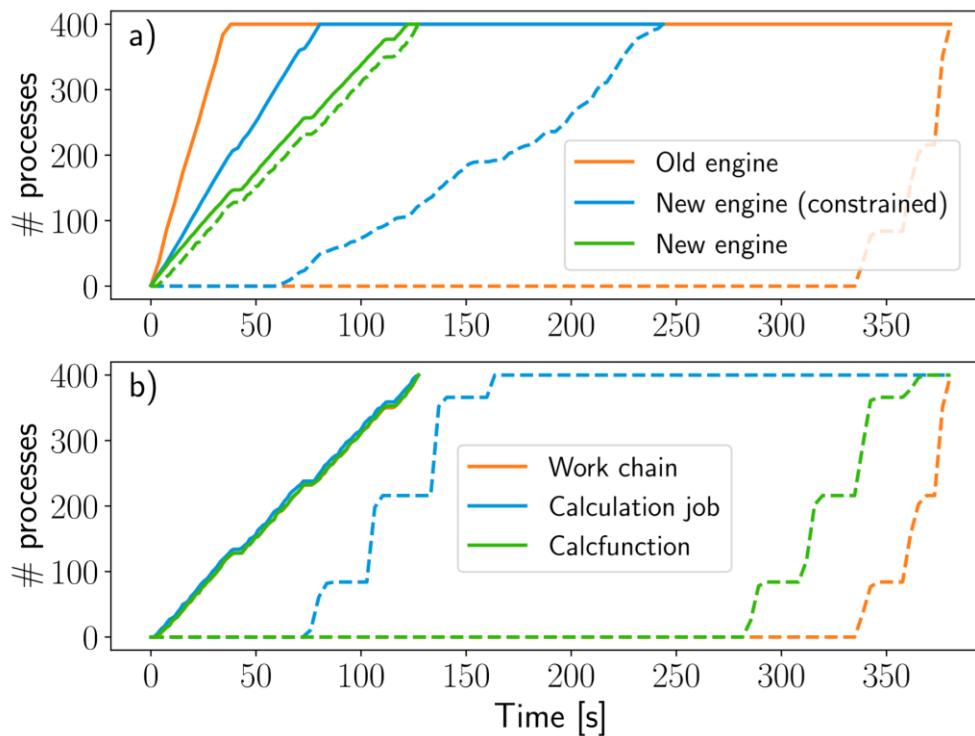


- Throughputs of **tens of thousands** of processes per hour

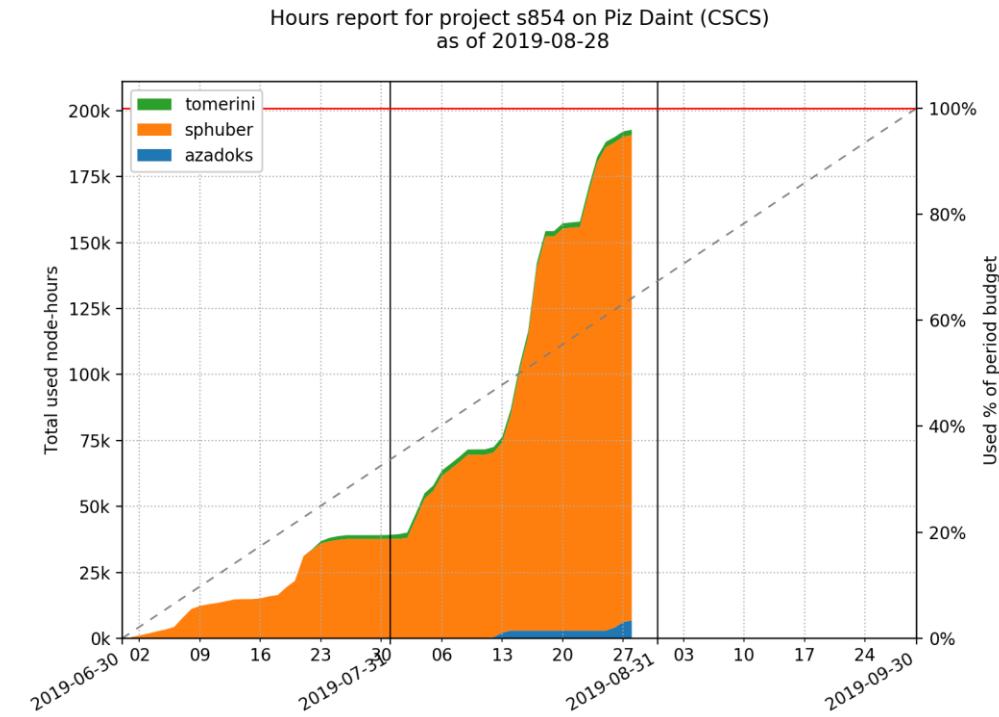
HIGH-THROUGHPUT TOWARDS THE EXASCALE

AiiDA 1.0: significant improvements to enable high-throughput computational loads [1]

- Responsive: changing from poll-based to event-based design
- Scalable: arbitrary number of independent daemon workers



- Easily fill the queue of multiple HPC resources
- Even for many small single-node jobs



- Throughputs of **tens of thousands** of processes per hour

- Sophisticated mechanisms to automatically deal with connection loss and bundling of requests to not overload the remote machines and schedulers

CONCLUSIONS



- High-throughput automated calculations on normal computational or HPC infrastructure
- Reproducible and robust workflows with automated error-handling
- Automated tracking of full data and workflow provenance
- Queryable database for high-performance data-analytics
- Domain specific library for plane-wave electronic structure codes
- Interface to Quantum ESPRESSO
- Negligible differences in total energies, forces and stress compared to normal QE
- Average speed-up of factor 6

ACKNOWLEDGMENTS AND ONLINE RESOURCES



European Centre of Excellence MaX



École Polytechnique Fédérale de Lausanne



MARVEL National Centre for Competency in Research



Platform for Advanced Scientific Computing



WEBSITE
<http://www.aiida.net>



DOCUMENTATION
<https://aiida.readthedocs.io>



SOURCE CODE
github.com/aiidateam/aiida-core



MATERIALS CLOUD
<https://materialscloud.org>



Sebastiaan P. Huber^{a,b}, Spyros Zoupanos^{a,b}, Martin Uhrin^{a,b}, Leopold Talirz^{a,b}, Leonid Kahle^{a,b}, Rico Häuselmann^{a,b}, Dominik Gresch^c, Tiziano Müller^d, Aliaksandr V. Yakutovich^{a,b}, Casper W. Andersen^{a,b}, Francisco F. Ramirez^{a,b}, Carl S. Adorf^{a,b}, Snehal Kumbhar^{a,b}, Elsa Passaro^{a,b}, Conrad Johnston^{a,b}, Nicolas Mounet^{a,b}, Nicola Marzari^{a,b} and Giovanni Pizzi^{a,b}

[a] National Centre for Computational Design and Discovery of Novel (MARVEL)

[b] Theory and Simulation of Materials, École Polytechnique Fédérale de Lausanne (EPFL)

[c] Laboratory of Molecular Simulation, École Polytechnique Fédérale de Lausanne (EPFL)

[d] Institute for Theoretical Physics, ETH Zürich, 8093 Zürich, Switzerland

[e] Institut für Physikalische Chemie, University of Zürich

We would like to thank the following people for code contributions, bug fixes, improvements of the documentation, and useful discussions and suggestions:

Oscar Arbelaez, Michael Atambo, Valentin Bersier, Marco Borelli, Jocelyn Boullier, Jens Bröder, Ivano E. Castelli, Andrea Cepellotti, Keija Cui, Vladimir Dikan, Marco Dorigo, Y.-W. Fang, Fernando Gargiulo, Marco Gibertini, Daniel Hollas, Eric Hontz, Jianxing Huang, Espen Flage Larsen, Boris Kozinsky, Ian Lee, Daniel Marchand, Antimo Marrazzo, Andrius Merkys, Simon Pintarelli, Gianluca Prandini, Philipp Rüßmann, Philippe Schwaller, Ole Schütt, Chris Sewell, Andreas Stamminger, Atsushi Togo, Daniele Tomerini, Nicola Varini, Jason Yu, Austin Zadoks, Bonan Zhu and Mario Zic.



DRIVING THE EXASCALE TRANSITION

Follow us on:

THANKS

company/max-centre/

@max_center2

<http://www.max-centre.eu/>