Highly scalable data management with signac

Carl Simon Adorf, Paul M. Dodd, Sharon C. Glotzer

MICDE Symposium for the Center for Network and Storage Enabled Collaborative Computational Science
May 19\textsuperscript{th} 2017, Ann Arbor, MI
The **signac** Framework

- Open-source – *BSD 3 License*
- Freely available
- Implemented and tested for
  - Python 2.7.x/3.3+
  - PyPy and PyPy3
- Website: [glotzerlab.engin.umich.edu/signac](http://glotzerlab.engin.umich.edu/signac)
Why signac?

Paul Signac (1863-1935) and George Seurat (1859-1891) invented the Pointillist painting technique.

Cassis, Cap Lombard, Opus 196
The Primary Goals

1. Make data searchable and thus sharable to improve collaboration on data reuse of data.

2. Reduce effort required for the implementation of computational investigations.
A Basic Workflow Model

Requirements
1. reproducible
2. flexible
3. scalable

Define Parameter Space → Data Acquisition → Data Curation → Analysis → Publish!

signac

data + meta data

work flow

data flow
The Two Dimensions of Scalability
Commonly encountered challenges
(or lessons learnt...)

A data management system is hard to apply, when it

1. convolutes data and workflow management;
   *hard to apply to existing workflows*

2. requires running of extra services;
   *sometimes difficult to implement in HPC environments*

3. does not scale well;
   *high barrier of entry/ difficult transition from laptop to HPC environment*

4. is too complex or nontransparent.
   *high barrier of entry/ over-engineered/ difficult to adjust*
The signac Design Philosophy

Development is guided by the following set of principles:

1. Data and workflow management are independent
   *Concept-wise and implementation-wise*

2. No requirements besides Python
   *Provided by any modern HPC environment.*

3. Data Management is decentralized and local
   *Performance is largely determined by file system.*

4. Simple and transparent data model
   *The data storage model is based on document-oriented metadata and files.*
The **signac** Data Space

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>{a = 3.0, b = 5.00, c = x}</td>
</tr>
<tr>
<td>{a = 5.0, b = 2.75, c = x}</td>
</tr>
<tr>
<td>{a = 2.0, b = 2.00, c = x}</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

Parameter & Data Space
The signac Data Model

• Metadata
  – Document-oriented encoded in JSON
  – Divided into
    • state point: static data address (provenance)
    • document: dynamic meta data

• Data
  – File-oriented

1) Also used to store lightweight data.
Where exactly does **signac** fit in?

**my project/**

```
src/
```

```
data/
```

```
doc/
```

```
README.txt
```

... 

Wilson, Bryan, Cranston, Kitzes, Nederbragt, and Teal, 
Demonstration of Project Setup
Demonstration – Ideal Gas Law

The Ideal Gas Law

\[ pV = Nk_B T \]
Data Access

• Selection and iteration over data (sub) spaces

```python
for job in project.find_jobs({'p': 10.0}):
    print(job.statepoint())

{'p': 10.0, 'kT': 1.0, 'N': 1000}
```

• Integration with pandas data frames

```python
df[(df.fluid=="argon") & (df.p > 2.0) & (df.p <= 5.0)].V_gas.mean()
```
Data Storage Layout

<< interface >>
Project

manages

generates

/workspace
/data/my_project

Index
0385...
2215...
....

data
operation

{state point}

03585...
22a51...
ee617...

data
Manage an Index with MongoDB
Index Operation Demonstration
1. The project is associated with a data set.
2. We operate on the project's data space with data operations.
3. We use signac for data management, which utilizes an internal data index.
4. We can access the index directly to access metadata and data.
Master Indexes

Like any other index, a master index can be used for data access.

Master Index

...{
  "_id": "abc123...",
  "root": "/data/project_a/", ...
}

...{
  "_id": "def456...",
  "root": "/data/project_c/", ...
}
Main Capabilities

These goals were translated into the following capabilities:

1. File and metadata management on the file system.
2. Capability to search and select data by metadata.
3. Generation of data space (deep) indexes.
4. Export of metadata and data into external databases.
5. Provision of base workflows integrated with cluster scheduling systems.
Documentation

signac.readthedocs.io

Interactive jupyter Tutorial
Future Development

- Release of the next revision of **signac-flow**
- Better documentation and support for **non-python-driven workflows**
- Implementation of **synchronization** routines/recipes (scp, rsync, globus, etc.)
Thank you very much!

signac - simple data management

The signac framework aids in the management of large and heterogeneous data spaces.

The signac framework supports researchers in managing project-related data with a well-defined indexable storage layout for data and metadata. This streamlines post-processing and analysis and makes data collectively accessible.

signac is open-source and freely available for Python versions 2.7.x and 3.4+.

arXiv:1611.03543 [cs.DB]
SUPPORTING INFORMATION
The Big Picture

Data Operation

Indexing

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>{a = 3.0, b = 5.00, c = x}</td>
</tr>
<tr>
<td>{a = 5.0, b = 2.75, c = x}</td>
</tr>
<tr>
<td>{a = 2.0, b = 2.00, c = x}</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
import signac
project = signac.get_project()
for p in 0.1, 1.0, 10.0:
    sp = {'p': p, 'T': 1.0, 'N': 1000}
    with project.open_job(sp) as job:
        if 'V' not in job.document:
            V = sp['N'] * sp['T'] / sp['p']
            job.document['V'] = V

for doc in project.index():
    print(doc)