

Building a code independent database

Fawzi Mohamed

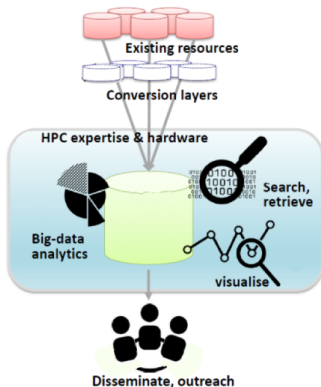
10.4.2016

- ▶ EU project started on the 1. Nov 2015
- ▶ 8 research centers
- ▶ 4 super computing centers
- ▶ 20 positions for highly qualified PhD students and postdocs



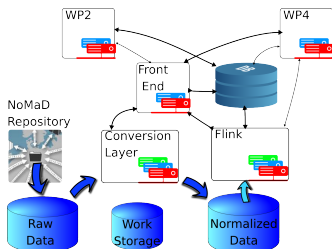
NOMAD Laboratory

- ▶ Start with open access calculations
- ▶ Data preparation (conversion layer)
- ▶ Big data analysis (insights)
- ▶ Visualization
- ▶ Materials Encyclopedia



Conversion layer and NOMAD Archive

- ▶ data preparation step
- ▶ Start with public data from the NOMAD repository
<http://nomad-repository.eu/>
- ▶ start from many codes and go to a code independent representation
- ▶ enable big data analysis between and across different codes



Big Data

- ▶ Big data in the data preparation phase is not primarily big amounts of data...
- ▶ ...but rather robust tools and work-flows.
- ▶ not just produce big amounts of data, but do so
- ▶ → in a reliable way
- ▶ → in a way that one can find problems, fix them
- ▶ ... and then integrate the correct data.

Naming and Provenance

- ▶ very important to achieve these goals
- ▶ about source of the data (a file,...)
- ▶ to describe the data itself

Description of a data source...

- ▶ public data from the repository
`http://nomad-repository.eu/`
- ▶ create raw data archives using the BagIt format
- ▶ use a name depending only on their content (modification dates filename and content of the data)
- ▶ → archiveGid: “R” + a 28-character checksum
- ▶ storage independent uris:
`nmd://<archiveGid>/path/in/archive`
- ▶ calculationGid: built from the uri of the main file of the calculation
- ▶ clear naming of the program used to transform the data: git describe for versions

Description of the normalized files

- ▶ comes from archive X (raw data naming and nomad uri), and was produced parser Y version Z1
- ▶ So we can say: “replace (or compare) that data with the one using parser Y version Z2”

Description of the data

- ▶ we want to describe workflows and changes on the data itself
- ▶ We use metadata as our conceptual model of the data
- ▶ Metadata is the name or label that characterizes corresponding values. For example, “fcc lattice constant in nm” is a metadata and “0.526” may be the corresponding value. Thus, if one thinks of storing data as ‘key’–‘value’ pairs (as in a dictionary), the ‘key’ is the metadata.
- ▶ metadata is boring
- ▶ allow seamless extension

Meta data: our conceptual model

- ▶ define how the data that we extract is organized, and what it is
- ▶ important both for human and for the machine
- ▶ data values consist of simple data types and multidimensional arrays of them
- ▶ group together similar *types* making them inherit from the same type (all energies inherit from the *energy*)
- ▶ group together *values* with sections
- ▶ allow one to many relationships between sections (relational data model)

Metadata: available now

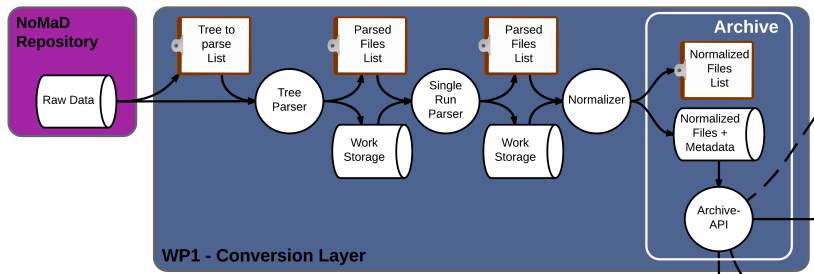
- ▶ common part: how to represent the common quantities found in calculations: energy, geometry, wavefunctions,...
- ▶ <https://nomad-dev.rz-berlin.mpg.de/ui/index.html>
- ▶
- ▶ open git, you can contribute!

Big data: speed and scalability

- ▶ Big Data is *also* about big amounts of data
- ▶ Optimization and being fast is important
- ▶ ability to scale, and use distributed resources

RabbitMQ, Kubernetes and Docker

- ▶ We use RabbitMQ as queuing system and docker container for the workers
- ▶ <http://kubernetes.io> is used to coordinate and scale the worker





- ▶ Flink, started in Berlin
- ▶ One of the leading frameworks for data-flow and streaming optimization that improves on the map reduce approach
- ▶ Tries to support not just data-flow or stream processing but also iterative methods, graph processing and some machine learning algorithms

Parsers...

- ▶ extract information from simulation input and outputs to make it available for analysis
- ▶ information that is not extracted is invisible, a parser defines the data that can be analyzed
- ▶ to make the data processable in an automatic way it should be mapped to a clear model
- ▶ a parser is a mapping from the inputs and outputs of a calculation to data described by our metadata

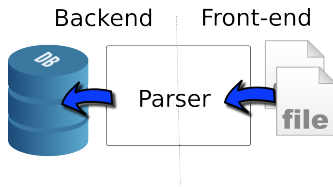
Many codes...

- ▶ FHI-aims , VASP , turbomole , Dmol³ , ORCA , TINKER , NAMD , exciting , WIEN2k , ELK , FLEUR , FPLO , Gaussian , GAMESS , NWChem , Molcas , GULP , onetep , CASTEP , LAMMPS , DL_POLY , LM Suite (TB-LMTO-ASA) , QUIP /libatoms/GAP , LAMMPS , cp2k , crystal , BigDFT , SIESTA , CPMD , Quantum Espresso , octopus , Smeagol , DFTB+ , abinit , GPAW , ASAP , gromacs , MOPAC
- ▶ ...many formats

What did we learn on parsers

- ▶ parsers should be fast because we want to apply them to large quantities of data (and re-parse regularly)
- ▶ parsers should be usable in various contexts
- ▶ code change in time, parsers need to evolve
- ▶ we will have maintain and improve them for a while

Decoupling the parsers



- ▶ Independent systems are more robust
- ▶ can be changed or optimized independently
- ▶ can be reused in different contexts
- ▶ but the interface has to be chosen carefully, because it will dictate performance and complexity

The ideal parser

- ▶ starts with a declarative parser capable of parsing basically all information contained in an output
- ▶ optimizes it to extract the quantities required to calculate the code independent representation
- ▶ calculate the code independent quantities and return them
- ▶ can be reused in different contexts
- ▶ we can later decide that a quantity we ignored is now of interest.
- ▶ NOMAD tries to deliver tools to make the creation development and testing of such parsers simpler

Doing analysis with parsed result

- ▶ the goal of all this is doing the rest of NOMAD, so analysis
- ▶ obviously after so much preparation things...
- ▶ ... failed ...
- ▶ well not really but some things were wrong, or missing
- ▶ I imagine you know how we went to fix them
- ▶ not working around them, but fixing them and improving their detection and correction
- ▶ big data means doing similar thing many times, so we will encounter these problem again, and it should be easier to get rid of them

Inspect and Debug

- ▶ some parsers did not end → now the parse executer can be queried about what it is doing, and when it started
- ▶ we collect detailed statistics on the parsing and create a DD that can be queried to compare different parsers, find regressions,...
- ▶ we have automatic tests (and we will need many more)
- ▶ we want to make all these tools available to who does the analysis
- ▶ normally he know what he will want to check

Big Data Analysis

- ▶ testing an working on a parser is already big data analysis
- ▶ we try to already use some of the techniques for Big data analysis already here
- ▶ exploring notebooks
- ▶ support the parser developer, he knows best how to test

Generating calculations

- ▶ computer get faster
- ▶ it is easier to analyze output if you control the generation, you can have simpler analysis
- ▶ you can know things implicitly: calculation A and be use compatible settings, because I generated like that
- ▶ NOMAD choose the more difficult path of try to help even analyzing calculations done by somebody else
- ▶ much more difficult, but
- ▶ produces more robust parsers
- ▶ everybody needs parsers, and gains from more robust parsers
- ▶ even if the idea of analyzing random inputs should fail

Conclusions (philosophy and marketing)

- ▶ one answer, 100 questions: create tools can be inspected, debugged, and detect issues early
- ▶ empower the user (in this context the parser user, but true in general), give him useful info to do his analysis
- ▶ → be open we cannot do everything: create tools to enable the work of others and collaborate in doing so
- ▶ NOMAD: building the tools for your next analysis
- ▶ not yet there, but hopefully on the right way ;)

Acknowledgements

- ▶ Luca Ghiringhelli
- ▶ Ankit Kariryaa
- ▶ the parser developers
- ▶ Matthias Scheffler and the whole NOMAD Team
- ▶ you for your attention
- ▶ H2020 NOMAD

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 676580.