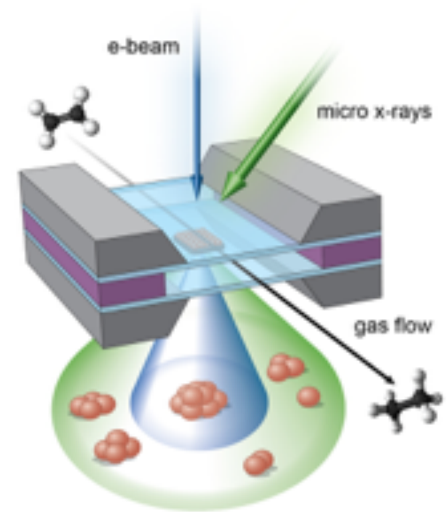
Reliable Performance forStreaming Analysis Workflows

BNL: Kerstin Kleese van Dam

SDSC: Ilkay Altintas

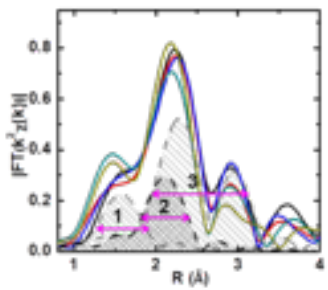
PNNL: Eric Stephan, Todd Elsethagen, Bibi Raju, Darren Kerbyson, Kevin Barker, Nathan Tallent, Jian Yin

Use Case: *In Operando* catalysis experiments



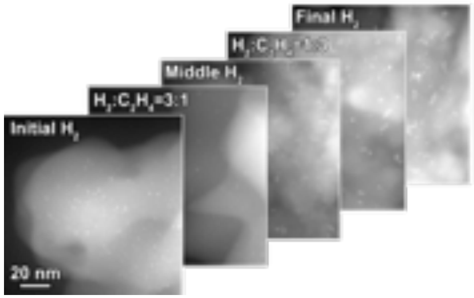
- Experimental measurements made with sample 'in a working condition'
- Different measurements needed to capture all aspect of system
- Multi—Modal, In-situ analysis coupled with predictive modeling transformative providing understanding and control of process

Data sets from different techniques:
Integration of data for highest scientific impact



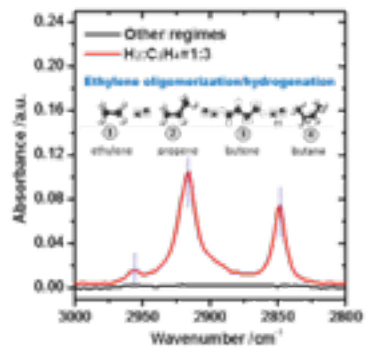
X-ray Absorption Spectroscopy

Global average structure and electronic structure



Transmission Electron Microscopy

Physical and electronic structure of individual catalysts



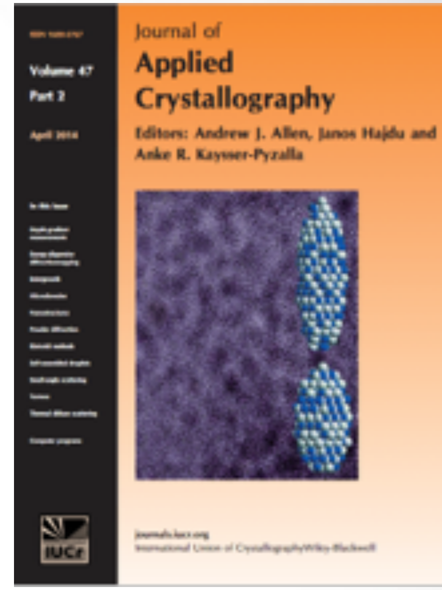
Infrared Spectroscopy

Direct determination of surface adsorbates

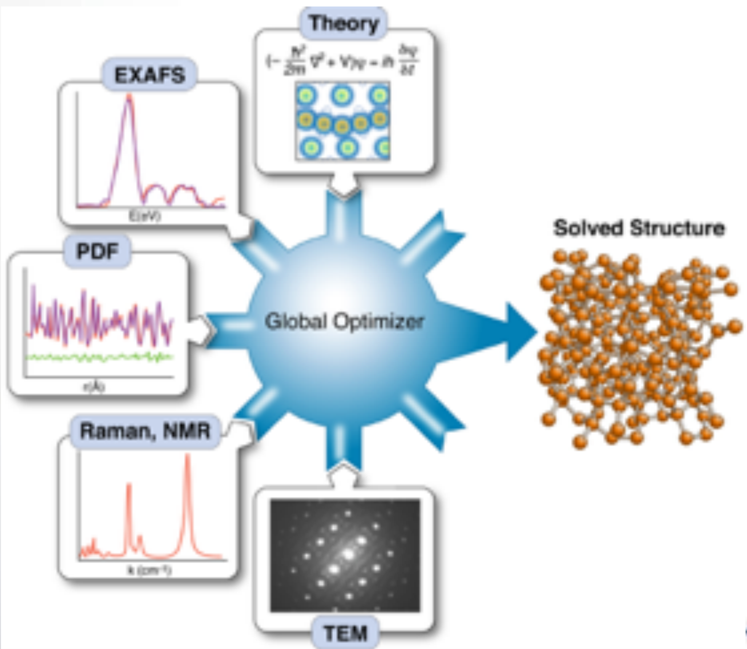
Stach, Frenkel
Nat. Comm. 2015

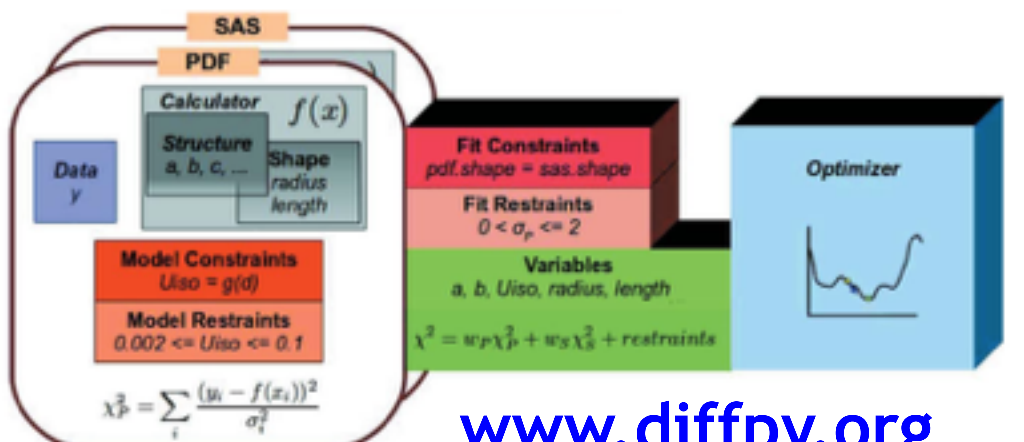
Complex Modeling

- Use of multiple data and information improves reliability by defining limits of both calculated and experimental results
- DiffPy-CMI, SumLib and SciKit-Beam in the CiffPy framework provide a streaming data integration and analysis framework for experimental and numerical simulation data.
- Many application use cases see web site.



Billinge, J. Appl. Cryst., 2014



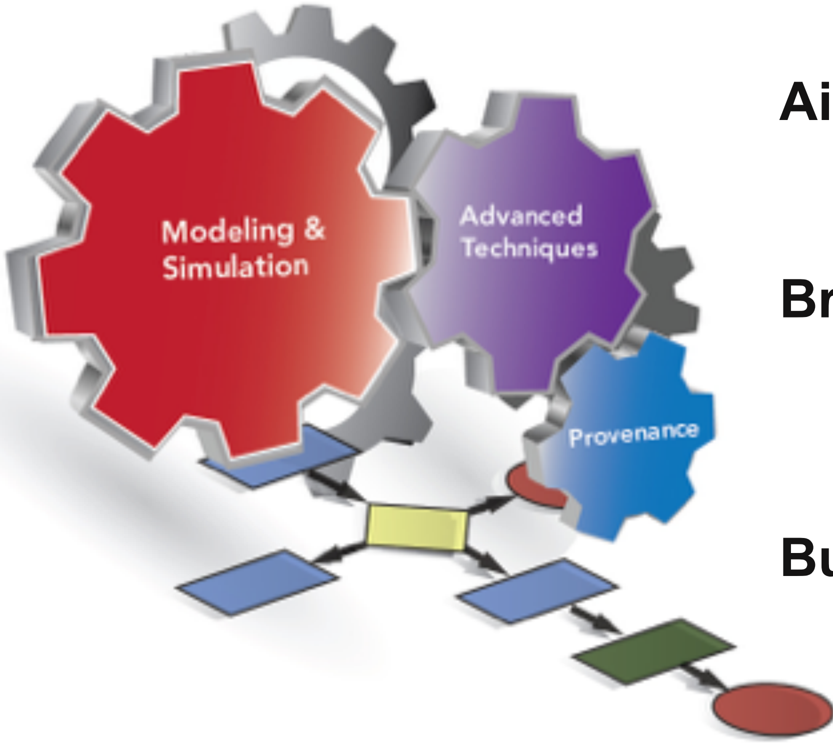


www.diffpy.org

Challenges in in-situ experimental analysis

- **Goal** - Provide enough targeted information to the scientists, early enough, to enable them to take critical decisions on steering of the data taking and its analysis
- **Critical characteristics**:
  - Speed, Accuracy, Completeness (incl. background, prediction)
  - Information selection and representation
  - Different programing languages, programming models, heterogenous data, computing and networking infrastructure
- **Essential - Reliable in Time Result Delivery**

DOE ASCR - Integrated End-to-End Performance Prediction and Diagnosis for Extreme Scientific Workflows



- Aim to provide an integrated approach to the modeling of extreme scale scientific workflows
- Brings together researchers working on modeling / simulation / empirical analysis, workflows and domain scientists
- Builds upon existing research much of which has focused to date on large-scale HPC systems and applications

Explore in advance - Design-space exploration & Sensitivity Analyses

Optimize at run-time - Guide execution based on dynamic behavior

Expanding Provenance: Empirical Information Gathering

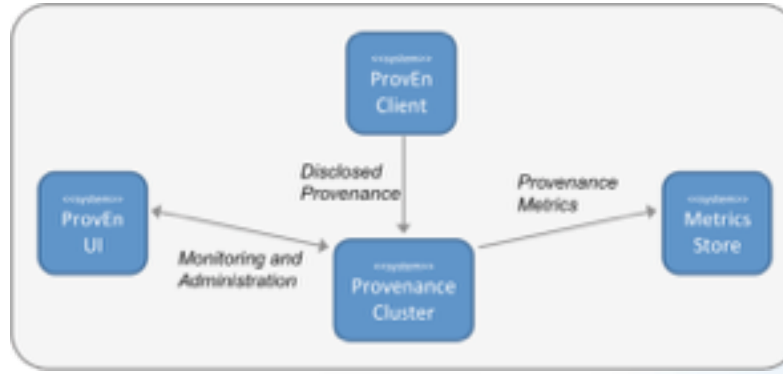
Today we only have hypothesis on what causes the variability in workflow performance or how performance could be improved

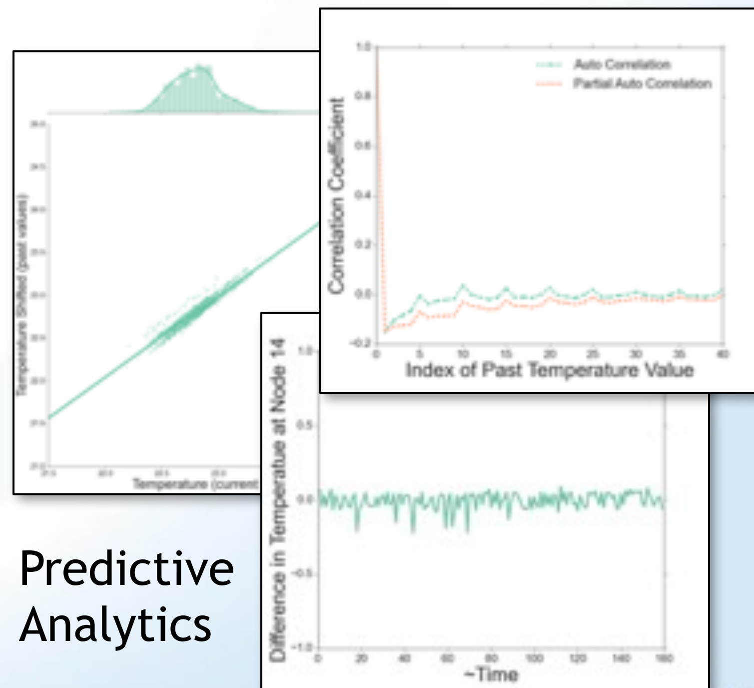
IPPD will use provenance to capture empirical performance information from workflows and systems to:

- Collect quantitative performance information to investigate workflow performance variability, degradation, sensitivity and impact
- Provide empirical data backed assessments of particularly prevalent performance bottlenecks and sources of performance variability
- Provide a record of performance changes over time that can be correlated with changes to applications, workflows and systems

ProvEn Overview

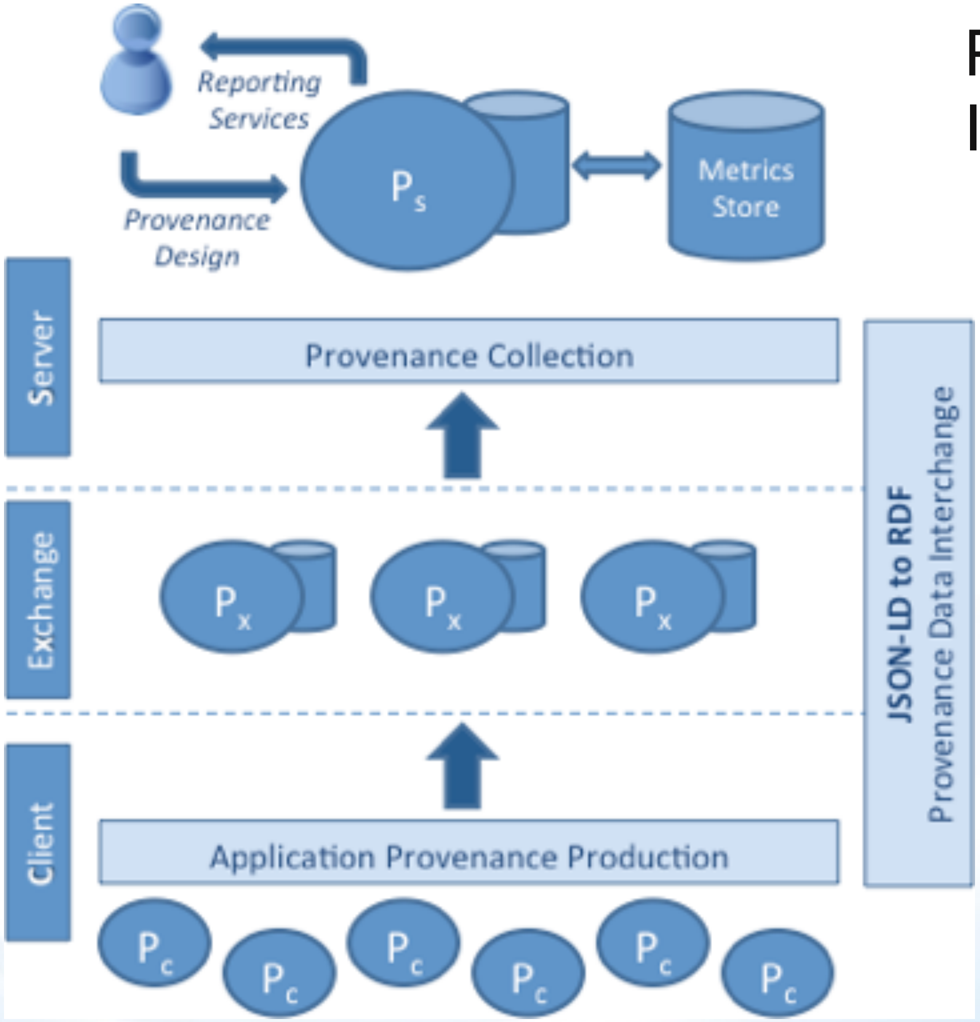
- Provenance Environment (ProvEn) - A Provenance production and collection framework.
- Provides services and libraries to collect provenance produced in a distributed environment
- ProvEn Client API aids in the production of provenance from client applications
- The following types of provenance are collected:
  - Time series-based information from a system/host perspective
  - Performance metrics tracking from an application/ workflow perspective
- ProvEn enables building of accurate Machine Learning models by capturing detailed footprints of large-scale execution traces.
- ProvEn will support identification of sources of performance variability in streaming analysis workflows, and provide runtime guidance to resource allocation systems.





Predictive Analytics

Provenance Environment (ProvEn) Architecture



ProvEn Services Infrastructure

- Provenance capture through messaging services and web service APIs
- Server / provenance consumer (semantic information, triple store)
- Client API library / provenance producer
- Time-series client/server (in progress, InfluxDB)

Initial System Test and Validation

Test System: SeaPearl at PNNL - 52 node cluster, instrumented with sensors that include temperature and power usage

Test Application: Firestarter, a stress test tool that can create varying workloads with predictable amounts of heat generation by the CPUs

Sampling Speed: Two nodes are monitored at 10KHz / 36M measurements / hour using a Lua script running on each node that pipes streaming measurements in parallel into the InfluxDB database.

Correlation: To correlate performance measures in the time series database to the provenance store the Network Time Protocol (NTP) is relied upon as the time source.

Questions?

Kerstin Kleese van Dam, kleese@bnl.gov