A framework for fully autonomous design of materials via multiobjective optimization and active learning

Tyler Chang[1], Jakob Elias[1], Stefan Wild[2], Santanu Chaudhuri[1,3] and Joseph Libera[1]

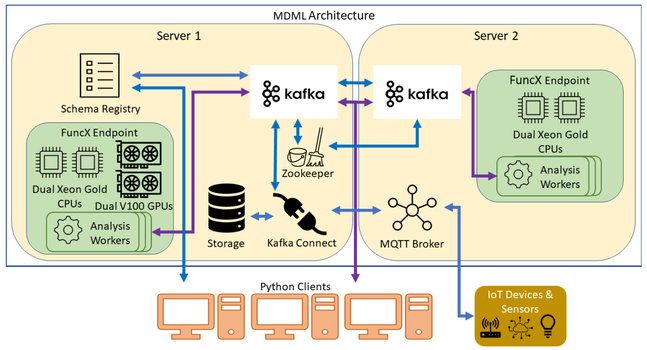
[1]*Argonne National Laboratory* [2]*Lawrence Berkeley National Laboratory* [3]*University of Illinois Chicago*

Our (Big) Goals

- Design a software framework for *self-driving labs*
- Accelerate discovery via intelligent experimentation
- Democratize lab-work by building open-source tools

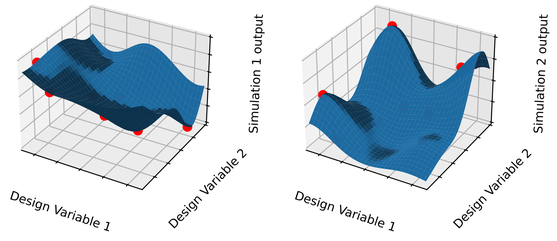
Streaming data from multiple sources

The *MDML* is a Kafka based platform for streaming, analyzing, and logging experiment and simulation data



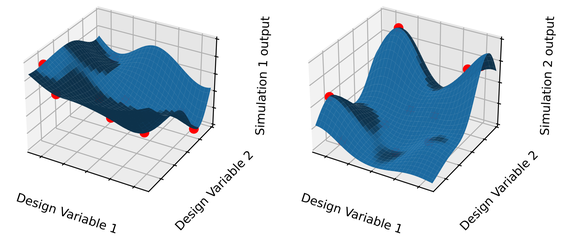
Model-based optimization

- **Search/sample** raw **experimental data**
- Use **surrogates** to model **experiments**
- Define **objectives + constraints** on **experiments**
- Use multiobjective **acquisition functions** to set targets
- **Solve acquisition on surrogates** ⇒ next **experiment**



Response Surface Methodology

- **Search/sample** data for raw **simulations outputs**
- Use **surrogates** to model **simulations**, **not objectives**
- Separately define **objectives and constraints**
- Scalarize **objectives** using **acquisition functions**
- **Solve scalarized surrogate problems** and iterate



Design Principles

Mix-and-match

- Initial search (design-of-experiments)
- Surrogate models
- Acquisition/scalarization functions
- Scalar optimization solvers

Easy for users and developers

- Support for variety of design vars and simulations
- Support various scientific workflows
- Embed/extract problems from unit cube

Flexible problem definitions

- Add design vars, sims, objs, + constraints
- Add searches, surrogates, acquisitions, optimizer
- Solve serially or in parallel using `libEnsemble`

Download ParMOO

- `git clone https://github.com/parmoo/parmoo`
- `pip install parmoo`

parMOO pypi

Continuing Work

- Continue to add new solvers and techniques
- Support wider variety of problems & workflows