

A web-based service for distributing curve fitting algorithms onto remote server clusters for improved performance

Raúl Negrón¹
Paul Kienzle²

¹University of Puerto Rico at Río Piedras

²NIST Center for Neutron Research

August 3, 2017



Disclaimer

Certain commercial equipment, instruments, or materials are identified in this presentation to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment are necessarily the best available for the purpose.

Project Idea

Bumps

Software package written in Python for curve fitting and uncertainty analysis.

- Build a web service back-end for running remote jobs
- Extend *bumps* functionality to integrate the web service

Project Goals

- Allow for improved performance

Project Goals

- Allow for improved performance
- Minimal setup for both server administrators and clients

Project Goals

- Allow for improved performance
- Minimal setup for both server administrators and clients
- Lower barrier to entry for interested users

Computer cluster A collection of interconnected computers (nodes) that perform computationally-intensive work

Terms

Computer cluster A collection of interconnected computers (nodes) that perform computationally-intensive work

HPC "High performance computing"; Used to describe servers built for performance ("Supercomputers")

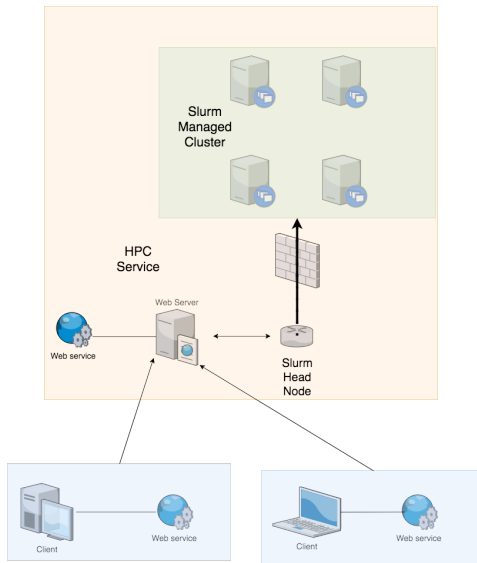
Terms

Computer cluster A collection of interconnected computers (nodes) that perform computationally-intensive work

HPC "High performance computing"; Used to describe servers built for performance ("Supercomputers")

Workload Manager Resource access control for a cluster. Common tasks include job scheduling and load management.

Service Diagram



Bumps basics

Optimizers include...

- Levenberg–Marquardt
- Nelder-Mead
- DREAM

Some other features

- Markov Chain Monte Carlo sampler
- Optional parallelism using MPI
- Built-in optional graphical interface

Bumps example (Linear Equation)

curve.py

```
from bumps.names import *

x = [1, 2, 3, 4, 5, 6]
y = [2.1, 4.0, 6.3, 8.03, 9.6, 11.9]
dy = [0.05, 0.05, 0.2, 0.05, 0.2, 0.2]

def line(x, m, b=0):
    return m * x + b

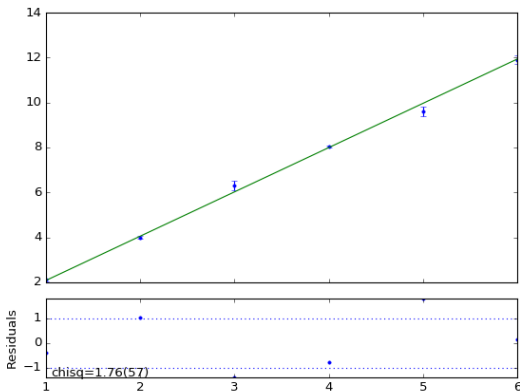
M = Curve(line, x, y, dy, m=2, b=2)
M.m.range(0, 4)
M.b.range(-5, 5)

problem = FitProblem(M)
```

Running the curve fit

```
$ bumps curve.py --fit=newton --steps=100 --store=temp
```

produces...



Bumps performance and a motivation for the web service

- Small problem space; efficient algorithms don't affect much
- Model functions can be somewhat expensive
- Running multiple models in parallel is an easy way to improve performance

Slurm



Slurm basics

- Free and open-source software
- Job scheduler
- Progress monitor
- Load manager
- Available on many modern server clusters (Ex. UMD)

Slurm example

```
job.slurm
```

```
#!/bin/bash
```

```
#SBATCH --mail-user=raul@upr.edu
```

```
#SBATCH --nodes=2
```

```
#SBATCH --ntasks=1
```

```
#SBATCH --time=00:10:00
```

```
#SBATCH --mem=500M
```

```
#SBATCH --output=python_job.out
```

```
python problem.py
```


Primary Services



Flask
web development,
one drop at a time



redis



docker

- A Flask server is used to implement the web service API
- Redis is used to store web service metadata such as:
 - Job-User association
 - Job submission date
- Docker provides service containers for easy setup

Docker and the Docker logo are trademarks of Docker, Inc.

Redis and the Redis logo are trademarks of Salvatore Sanfilippo

Docker based installation of the web service

```
$ docker pull redis:latest
```

```
$ docker build . -t bumps
```

```
$ docker run --name bumps_redis -d redis
```

```
$ docker run --link bumps_redis -p 5000:5000 bumps
```

Video Demo

`https://youtu.be/rcegYcqtREo`

`https://vimeo.com/226938525`

Landing page

NCNR bumps Fitting Test Server

Welcome. Please use your user token to sign in or request a new user token.

Enter your token:

Don't have access?

Note: This page currently uses cookies to remember your authorization token.

For help, please contact [support](#).

User dashboard before job submission

NCNR bumps Fitting Test Server

Logged in as token c68727.

You currently have no jobs running.

Submit a new job...

For help, please contact [support](#).

Setting up a bumps/slurm job

NCNR bumps Fitting Test Server

Specify job options

Limit this job to one node? ☐

Number of processor cores across all nodes:

64 

Number of GPUs:

Memory per processor core:

16

Memory Unit

GB 

Walltime (HH:MM:SS)

00:00:40

Jobname

My Very Important Job

steps:

burn:

Fit Optimizer: 

Model file: polisson.py

Email address:

User dashboard after job submission

- Job submitted successfully.

NCNR bumps Fitting Test Server

Logged in as token c68727.

[View the status of your jobs...](#)

Job #1 (Submitted on Mon Jul 24 14:08:41 2017) | Status: pending...

Submit a new job...

Results after job completion

NCNR bumps Fitting Test Server

Logged in as token c68727.

[View the status of your jobs...](#)

Job #1 (Submitted on Mon Jul 24 14:08:41 2017) | Results available!

File: poisson-chain.mc [Download](#)

File: poisson-point.mc [Download](#)

File: poisson-stats.mc [Download](#)

File: poisson.dat [Download](#)

File: poisson.err [Download](#)

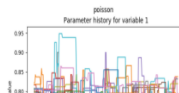
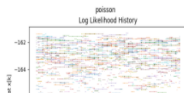
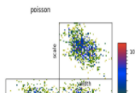
File: poisson.log [Download](#)

File: poisson.mon [Download](#)

File: poisson.par [Download](#)

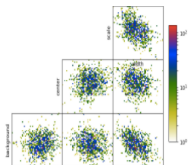
Download zipped results (excluding .mc files) [Download](#)

Generated graphs (click to interact)

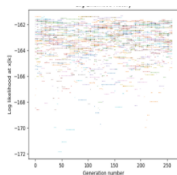


Web page example

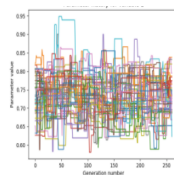
Results after job completion (cont.)



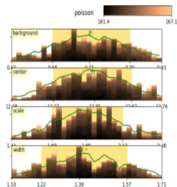
[Download .png file](#)



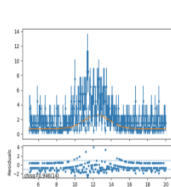
[Download .png file](#)



[Download .png file](#)



[Download .png file](#)

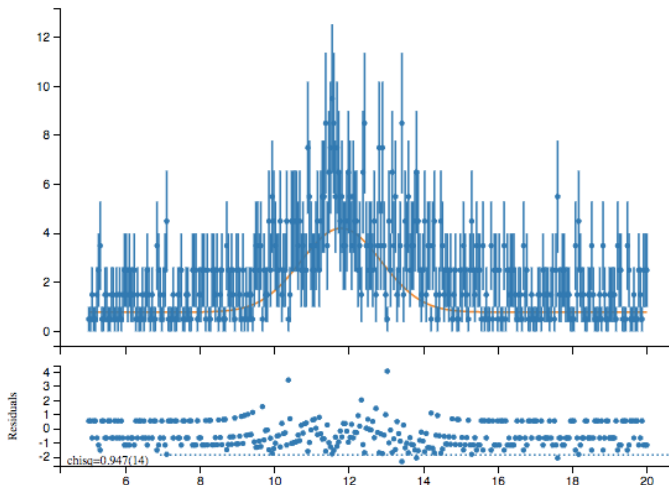


[Download .png file](#)

Delete job #1: [Delete](#)

Submit a new job... [Go](#)

Resulting Matplotlib/D3.js HTML model graph



Resulting Slurm file

```
#!/bin/bash

#SBATCH job-name=My_Very_Important_Job
#SBATCH --mail-user=raul@me.com
#SBATCH --time=00:00:40
#SBATCH --ntasks=64
#SBATCH --mem-per-cpu=16G

bumps [...]/[user_folder]/poisson.py \
    --batch --stepmon --burn=600 --fit=dream \
    --store=[...]/[user_folder]/results
```

Results

- Client/Server infrastructure using Flask
- Docker images for simple setup
- Slurm file parsing from web forms and command line
- Simple web page interface to the service
- MPLD3-based interactive, in-browser graphs

Get the code

Project repository: `https://github.com/rnegrn/bumps`

Public domain code!

Acknowledgments

A big *thank you* to:

Paul Kienzle, Julie Borchers, Joe Dura,
Brandi Toliver



Thank you!

Questions?