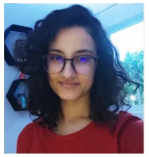
Update from BRA-LIN S4
PZ Services for LSST

Julia Gschwend

IDAC BRASIL
LSST

Julia Gschwend
Astronomer
PZ Contribution Lead

Carlos Adean
System Analyst
IDAC Contribution Lead



Heloisa Mengisztki
Undergrad student in
Computer Science

Cristiano Singulani
Back-end developer

Glauber Vila-Verde
Front-end developer

Jandson Vitorino
Front-end developer

Henrique Dante
Software engineer

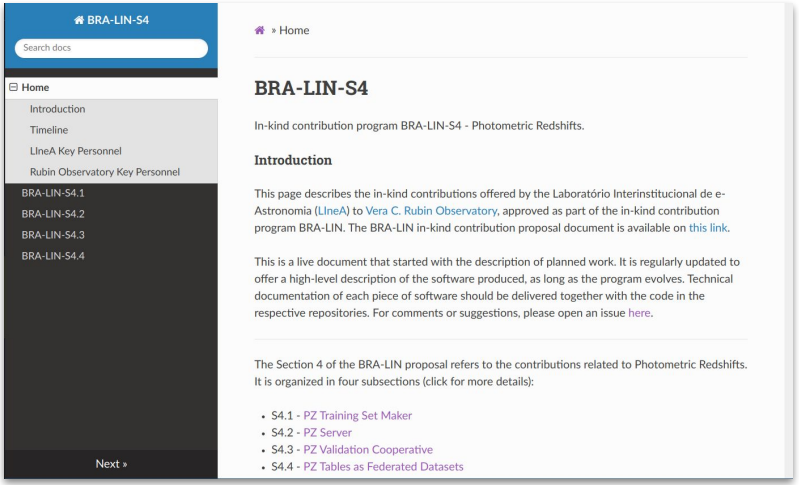
Attending PCW 2023 in person

BRA-LIN S4 - PZ Services

Overview

1. **S4.1 + S4.2: PZ Server**
2. **S4.4: PZ Compute**
3. **S4.3: PZ Validation Cooperative**

[Overview documentation page:
https://linea-it.github.io/pz-lsst-inkind-doc/](https://linea-it.github.io/pz-lsst-inkind-doc/)



LIneA

1. PZ Server

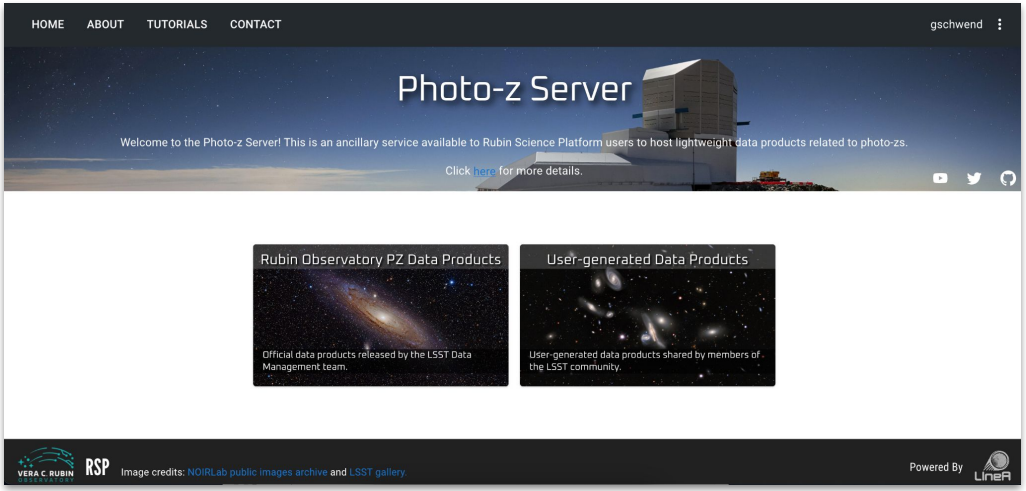
The website and data storage service

Advanced stage of development!
Test environment available at:
https://pz-server-dev.linea.org.br/

Repo: linea-it/pzserver_app
Project: PZ Server

Overview:

- Host service for lightweights pz-related data products (e.g., training sets).
- Separate pages for Rubin official and user-generated data products.

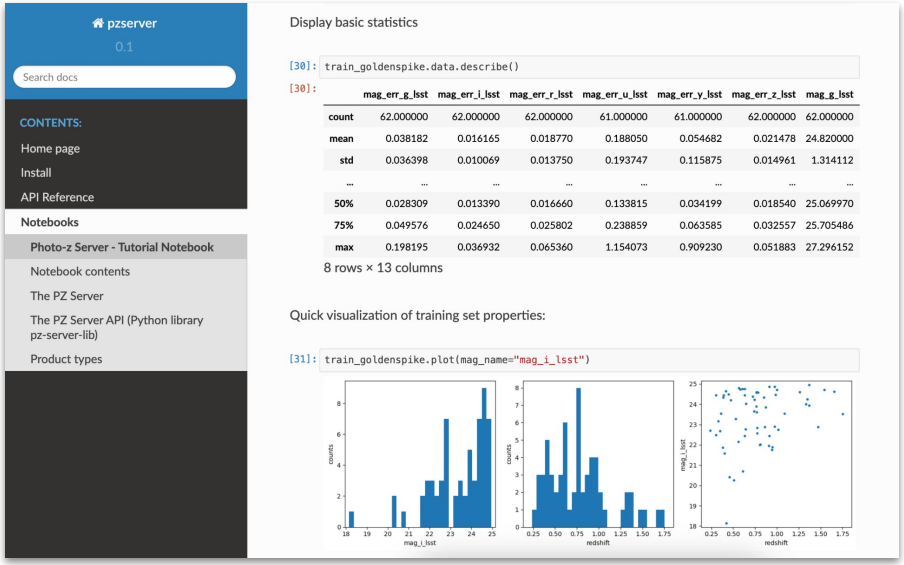


1. PZ Server

The website and data storage service

Advanced stage of development!
Test environment available at:
https://pz-server-dev.linea.org.br/

Repo: linea-it/pzserver_app
Project: PZ Server



Overview (cont.):

Python API

Library:

```
$ pip install pzserver
```

Tutorial notebook:

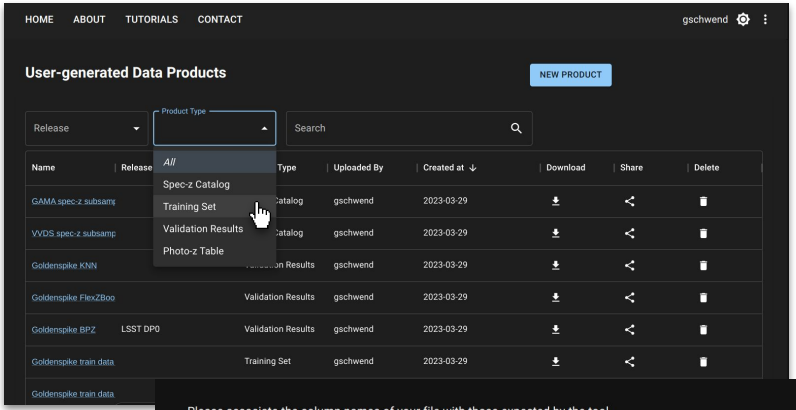
```
$ git clone https://github.com/linea-it/pzserver.git
```

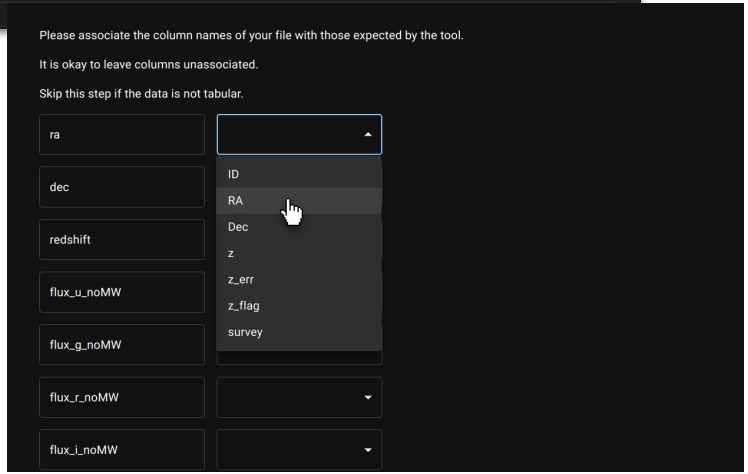
1. PZ Server

The website and data storage service

Repo: [linea-it/pzserver_app]

Project: [PZ Server]





New features since the PZ Symposium:

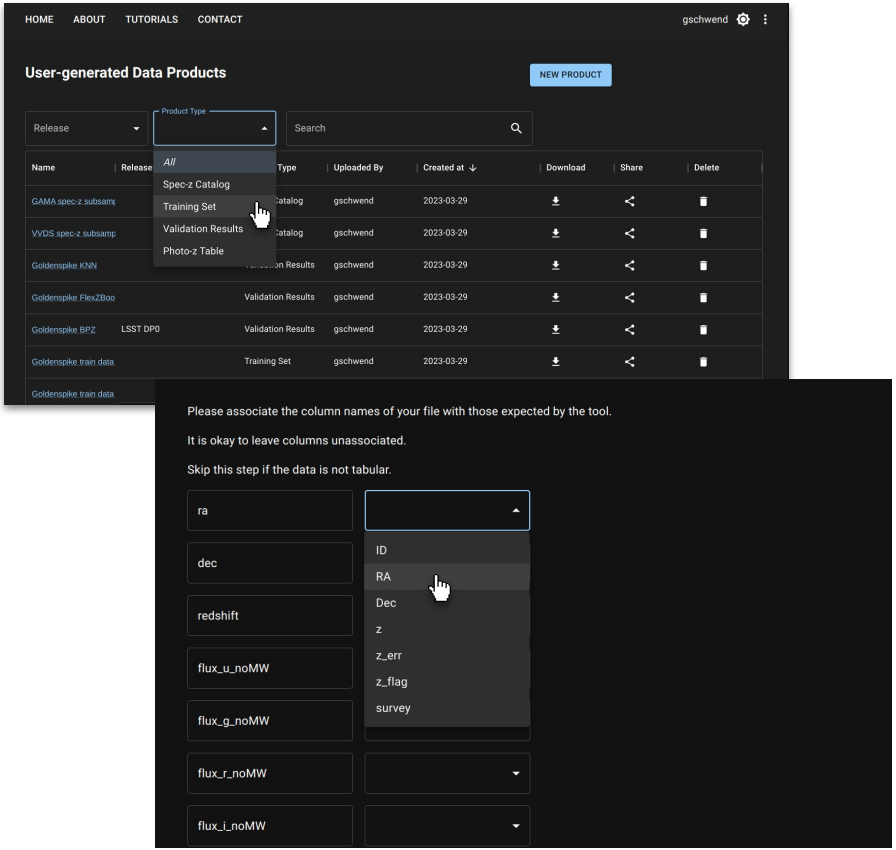
- Dark mode
- Share button
- Write-in inputs for column association (PR in review)

1. PZ Server

The website and data storage service

Repo: [linea-it/pzserver_app]

Project: [PZ Server]



Your feedback is welcome!

Test features and tell us about your user experience:

- Download data products
- Upload data products
- Share data products
- Access metadata via Jupyter notebook
- Access tabular data via Jupyter notebook

[subscription link]

[feedback form]

1. PZ Server

The *Training Set Maker* service

linea-it/pzserver

* Early stage of development.

Cross-matching service to create Training Sets on demand.

Back-end:

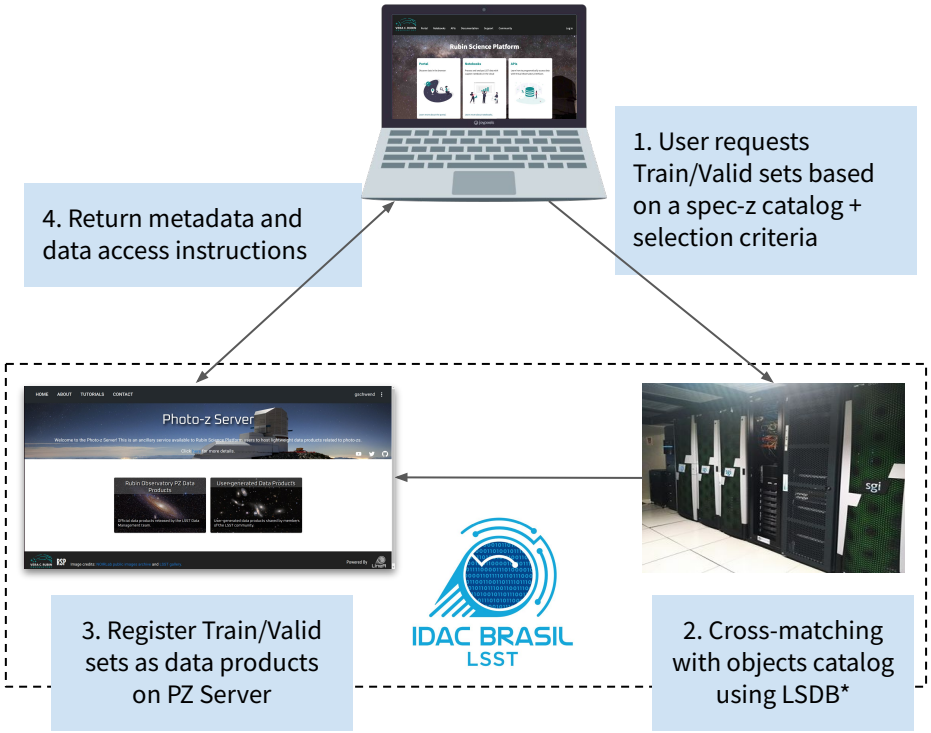
- Orchestration system to receive users requests and run processes at the IDAC
- Data partitioning based on HiPSCat
- LSDB as cross-matching tool

} developed by LINCC team

- Training Sets created become available as a PZ Server data product

Front-end :

- PZ Server API (methods in PZ Server class)
- Pipelines page on the PZ Server (NEW)



2. PZ Compute

PZ Tables as federated dataset

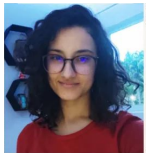
Repo: linea-it/pz-compute

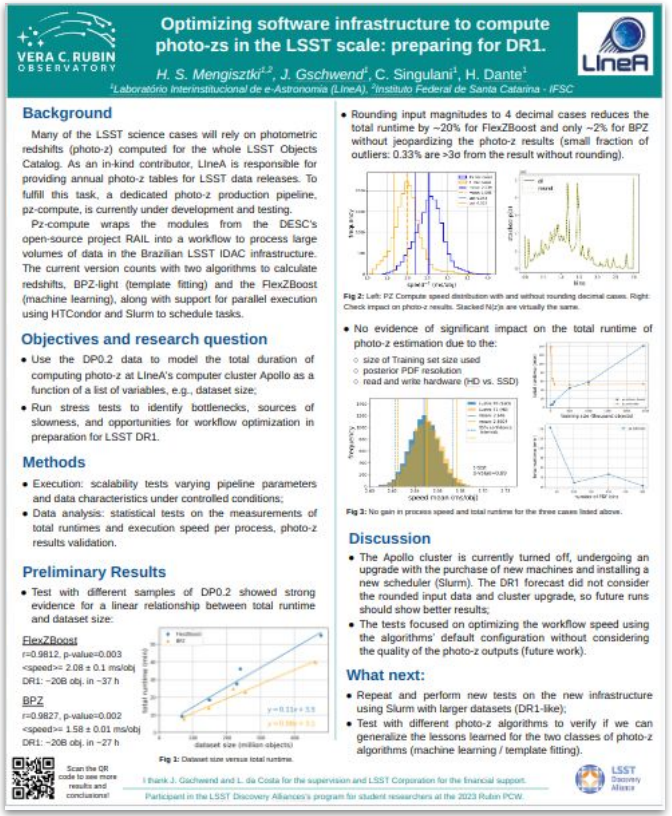
Project: [PZ Compute]

Overview:

- Pipeline to automatize the dispatch of jobs to:
  - LIneA's cluster Apollo
  - Supercomputer Santos Dumont
- Uses RAIL evaluation as PZ code wrapper
- Pre-process step
- Pipeline optimization study

Poster presentation tomorrow
by Heloisa Mengisztki
Undergrad student in
Computer Science





VERA C. RUBIN OBSERVATORY

Optimizing software infrastructure to compute photo-zs in the LSST scale: preparing for DR1.

LIneA

H. S. Mengisztki[1,2], J. Gschwend[2], C. Singulani[1], H. Dante[1]
[1]Laboratório Interinstitucional de e-Astronomia (LIneA), [2]Instituto Federal de Santa Catarina - IFSC

Background

Many of the LSST science cases will rely on photometric redshifts (photo-z) computed for the whole LSST Objects Catalog. As an in-kind contributor, LIneA is responsible for providing annual photo-z tables for LSST data releases. To fulfill this task, a dedicated photo-z production pipeline, pz-compute, is currently under development and testing.

Pz-compute wraps the modules from the DESC's open-source project RAIL into a workflow to process large volumes of data in the Brazilian LSST IDAC infrastructure. The current version counts with two algorithms to calculate redshifts, BPZ-light (template fitting) and the FlexZBoost (machine learning), along with support for parallel execution using HTCondor and Slurm to schedule tasks.

Objectives and research question

- Use the DP0.2 data to model the total duration of computing photo-z at LIneA's computer cluster Apollo as a function of a list of variables, e.g., dataset size;
- Run stress tests to identify bottlenecks, sources of slowness, and opportunities for workflow optimization in preparation for LSST DR1.

Methods

- Execution: scalability tests varying pipeline parameters and data characteristics under controlled conditions;
- Data analysis: statistical tests on the measurements of total runtimes and execution speed per process, photo-z results validation.

Preliminary Results

- Test with different samples of DP0.2 showed strong evidence for a linear relationship between total runtime and dataset size:

FlexZBoost
r=0.9812, p-value=0.003
<speed>= 2.08 ± 0.1 ms/obj
DR1: ~20B obj. in ~37 h

BPZ
r=0.9827, p-value=0.002
<speed>= 1.58 ± 0.01 ms/obj
DR1: ~20B obj. in ~27 h

Fig 1: Dataset size versus total runtime.

- Rounding input magnitudes to 4 decimal cases reduces the total runtime by ~20% for FlexZBoost and only ~2% for BPZ without jeopardizing the photo-z results (small fraction of outliers: 0.33% are >3σ from the result without rounding).

Fig 2: Left: PZ Compute speed distribution with and without rounding decimal cases. Right: Check impact on photo-z results. Stacked N(z)s are virtually the same.

- No evidence of significant impact on the total runtime of photo-z estimation due to the:
  - size of Training set size used
  - posterior PDF resolution
  - read and write hardware (HD vs. SSD)

Fig 3: No gain in process speed and total runtime for the three cases listed above.

Discussion

- The Apollo cluster is currently turned off, undergoing an upgrade with the purchase of new machines and installing a new scheduler (Slurm). The DR1 forecast did not consider the rounded input data and cluster upgrade, so future runs should show better results;
- The tests focused on optimizing the workflow speed using the algorithms' default configuration without considering the quality of the photo-z outputs (future work).

What next:

- Repeat and perform new tests on the new infrastructure using Slurm with larger datasets (DR1-like);
- Test with different photo-z algorithms to verify if we can generalize the lessons learned for the two classes of photo-z algorithms (machine learning / template fitting).

Scan the QR code to see more results and conclusions!

I thank J. Gschwend and L. da Costa for the supervision and LSST Corporation for the financial support.
Participant in the LSST Discovery Alliance's program for student researchers at the 2023 Rubin PCW.

LSST Discovery Alliance

LIneA

2. PZ Compute

PZ Tables as federated dataset

Repo: linea-it/pz-compute

Project: [PZ Compute]

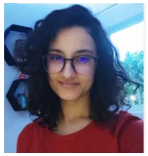
Overview:

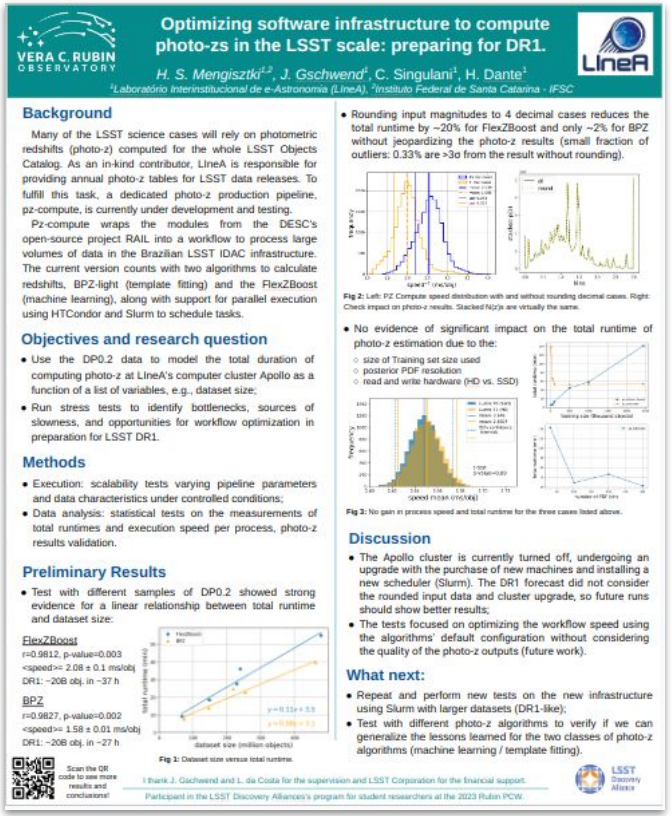
- Pipeline to automatize the dispatch of jobs to:
  - LIneA's cluster Apollo
  - Supercomputer Santos Dumont
- Uses RAIL evaluation as PZ code wrapper
- Pre-process step
- Pipeline optimization study

Forecast for DR1 (supposing objects catalog with 20Bi):

- Process time: < 2 days
- Storage space: ~45 TB (301 bins without compression)

Poster presentation tomorrow
by Heloisa Mengisztki
Undergrad student in
Computer Science





Optimizing software infrastructure to compute photo-zs in the LSST scale: preparing for DR1.

H. S. Mengisztki[1,2], J. Gschwend[2], C. Singulani[1], H. Dante[1]

[1]Laboratório Interinstitucional de e-Astronomia (LIneA), [2]Instituto Federal de Santa Catarina - IFSC

Background

Many of the LSST science cases will rely on photometric redshifts (photo-z) computed for the whole LSST Objects Catalog. As an in-kind contributor, LIneA is responsible for providing annual photo-z tables for LSST data releases. To fulfill this task, a dedicated photo-z production pipeline, pz-compute, is currently under development and testing.

Pz-compute wraps the modules from the DESC's open-source project RAIL into a workflow to process large volumes of data in the Brazilian LSST IDAC infrastructure. The current version counts with two algorithms to calculate redshifts, BPZ-light (template fitting) and the FlexZBoost (machine learning), along with support for parallel execution using HTCondor and Slurm to schedule tasks.

Objectives and research question

- Use the DP0.2 data to model the total duration of computing photo-z at LIneA's computer cluster Apollo as a function of a list of variables, e.g., dataset size;
- Run stress tests to identify bottlenecks, sources of slowness, and opportunities for workflow optimization in preparation for LSST DR1.

Methods

- Execution: scalability tests varying pipeline parameters and data characteristics under controlled conditions;
- Data analysis: statistical tests on the measurements of total runtimes and execution speed per process, photo-z results validation.

Preliminary Results

- Test with different samples of DP0.2 showed strong evidence for a linear relationship between total runtime and dataset size:

FlexZBoost
r=0.9812, p-value=0.003
<speed>= 2.08 ± 0.1 ms/obj
DR1: ~20B obj. in ~37 h

BPZ
r=0.9827, p-value=0.002
<speed>= 1.58 ± 0.01 ms/obj
DR1: ~20B obj. in ~27 h

Fig 1: Dataset size versus total runtime.

Scan the QR code to see more results and conclusions!

- Rounding input magnitudes to 4 decimal cases reduces the total runtime by ~20% for FlexZBoost and only ~2% for BPZ without jeopardizing the photo-z results (small fraction of outliers: 0.33% are >3σ from the result without rounding).

Fig 2: Left: PZ Compute speed distribution with and without rounding decimal cases. Right: Check impact on photo-z results. Stacked N(z)s are virtually the same.

- No evidence of significant impact on the total runtime of photo-z estimation due to the:
  - size of Training set size used
  - posterior PDF resolution
  - read and write hardware (HD vs. SSD)

Fig 3: No gain in process speed and total runtime for the three cases listed above.

Discussion

- The Apollo cluster is currently turned off, undergoing an upgrade with the purchase of new machines and installing a new scheduler (Slurm). The DR1 forecast did not consider the rounded input data and cluster upgrade, so future runs should show better results;
- The tests focused on optimizing the workflow speed using the algorithms' default configuration without considering the quality of the photo-z outputs (future work).

What next:

- Repeat and perform new tests on the new infrastructure using Slurm with larger datasets (DR1-like);
- Test with different photo-z algorithms to verify if we can generalize the lessons learned for the two classes of photo-z algorithms (machine learning / template fitting).

I thank J. Gschwend and L. da Costa for the supervision and LSST Corporation for the financial support.
Participant in the LSST Discovery Alliance's program for student researchers at the 2023 Rubin PCW.

2. PZ Compute

PZ Tables as federated dataset

Repo: linea-it/pz-compute

Project: [PZ Compute]

Overview:

- Pipeline to automatize the dispatch of jobs to:
  - LIneA's cluster Apollo
  - Supercomputer Santos Dumont
- Uses RAIL evaluation as PZ code wrapper
- Pre-process step
- Pipeline optimization study

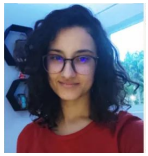
Forecast for DR1 (supposing objects catalog with 20Bi):

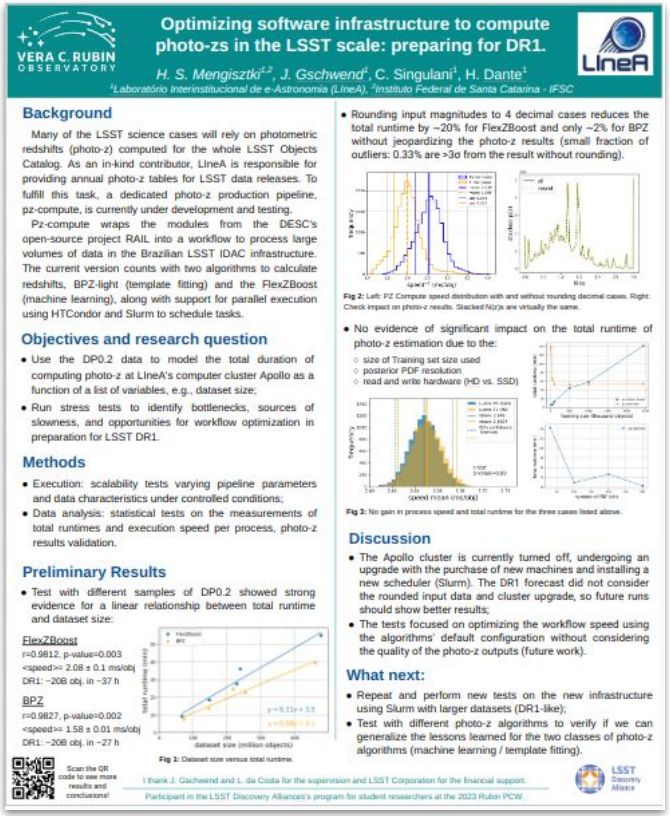
- Process time: < 2 days
- Storage space: ~45 TB (301 bins without compression)

Next steps:

- Scientific validation of DP0.2 PZ tables
- Upload DP0.2 PZ tables to USDF

Poster presentation tomorrow
by Heloisa Mengisztki
Undergrad student in
Computer Science





VERA C. RUBIN OBSERVATORY

Optimizing software infrastructure to compute photo-zs in the LSST scale: preparing for DR1.

LIneA

H. S. Mengisztki[1,2], J. Gschwend[2], C. Singulani[1], H. Dante[1]
[1]Laboratório Interinstitucional de e-Astronomia (LIneA), [2]Instituto Federal de Santa Catarina - IFSC

Background

Many of the LSST science cases will rely on photometric redshifts (photo-z) computed for the whole LSST Objects Catalog. As an in-kind contributor, LIneA is responsible for providing annual photo-z tables for LSST data releases. To fulfill this task, a dedicated photo-z production pipeline, pz-compute, is currently under development and testing.

Pz-compute wraps the modules from the DESC's open-source project RAIL into a workflow to process large volumes of data in the Brazilian LSST IDAC infrastructure. The current version counts with two algorithms to calculate redshifts, BPZ-light (template fitting) and the FlexZBoost (machine learning), along with support for parallel execution using HTCondor and Slurm to schedule tasks.

Objectives and research question

- Use the DP0.2 data to model the total duration of computing photo-z at LIneA's computer cluster Apollo as a function of a list of variables, e.g., dataset size;
- Run stress tests to identify bottlenecks, sources of slowness, and opportunities for workflow optimization in preparation for LSST DR1.

Methods

- Execution: scalability tests varying pipeline parameters and data characteristics under controlled conditions;
- Data analysis: statistical tests on the measurements of total runtimes and execution speed per process, photo-z results validation.

Preliminary Results

- Test with different samples of DP0.2 showed strong evidence for a linear relationship between total runtime and dataset size:

FlexZBoost
r=0.9812, p-value=0.003
<speed>= 2.08 ± 0.1 ms/obj
DR1: ~20B obj. in ~37 h

BPZ
r=0.9827, p-value=0.002
<speed>= 1.58 ± 0.01 ms/obj
DR1: ~20B obj. in ~27 h

Fig 1: Dataset size versus total runtime.

Scan the QR code to see more results and conclusions!

- Rounding input magnitudes to 4 decimal cases reduces the total runtime by ~20% for FlexZBoost and only ~2% for BPZ without jeopardizing the photo-z results (small fraction of outliers: 0.33% are >3σ from the result without rounding).

Fig 2: Left: PZ Compute speed distribution with and without rounding decimal cases. Right: Check impact on photo-z results. Stacked N(z)s are virtually the same.

- No evidence of significant impact on the total runtime of photo-z estimation due to the:
  - size of Training set size used
  - posterior PDF resolution
  - read and write hardware (HD vs. SSD)

Fig 3: No gain in process speed and total runtime for the three cases listed above.

Discussion

- The Apollo cluster is currently turned off, undergoing an upgrade with the purchase of new machines and installing a new scheduler (Slurm). The DR1 forecast did not consider the rounded input data and cluster upgrade, so future runs should show better results;
- The tests focused on optimizing the workflow speed using the algorithms' default configuration without considering the quality of the photo-z outputs (future work).

What next:

- Repeat and perform new tests on the new infrastructure using Slurm with larger datasets (DR1-like);
- Test with different photo-z algorithms to verify if we can generalize the lessons learned for the two classes of photo-z algorithms (machine learning / template fitting).

I thank J. Gschwend and L. da Costa for the supervision and LSST Corporation for the financial support.
Participant in the LSST Discovery Alliances's program for student researchers at the 2023 Rubin PCW.

LSST Discovery Alliance

LIneA

Thanks!

Any questions?

2. PZ Compute

PZ Tables as federated dataset

Repo: [linea-it/pz-compute]

Project: [PZ Compute]

Summary of PZ Compute stages (details [here]):

Stage I - Data acquisition
- Data transfer from USDF
- Organize/split into data partitions, data cleaning
- Download and store ancillary files

Stage II - Photo-z pre-processing
- Training PZ machine learning methods
- PZ scientific validation

Stage III - Photo-z computing
- Execute PZ codes (use RAIL as code wrapper)
- Validate results with Rubin PZ Coord.

Stage IV - Photo-z post-processing
- Data transfer from BR IDAC to USDF
- Register on RSP as a federated dataset.