An Update on Formats and Tools for Distributed Analyses of Large Datasets

Mario Juric
DiRAC Institute Director | LINCC FW
Professor of Astronomy, University of Washington

with Wilson Bebe, Doug Branton, Sandro Campos, Neven Caplar, Melissa DeLucchi, Jeremy Kubica, Kostya Malanchev, Sean McGuire, Colin Slater, Steven Stetzler, Max West, Sam Wyatt

and the LINCC Frameworks Analytics Group

DATA INTENSIVE RESEARCH IN ASTROPHYSICS AND COSMOLOGY

AST-2003196

The LINCC Frameworks Project

LSST Interdisciplinary Network For Collaboration And Computing

To collaboratively develop open computing systems and algorithms needed for large-survey analyses.

LSST Discovery Alliance

SCHMIDT FUTURES

Two LINCC-FW hubs:

- Carnegie Mellon University
- University of Washington

The Legacy Survey of Space and Time

Deep synoptic optical survey, coming in 2025.

Repeated imaging of the visible sky to ~24th mag.
10 years of operation.
60 PB of raw data.
40 billion stars, galaxies, asteroids.
30 trillion observations.

Scale of the problem

Rubin Year #1 dataset:

- 10Bn objects
- 100 obsv/object == 1T observations
- 100 bytes/obsv == **100 TB**

Not just a Rubin Problem:

- Gaia, DES, ZTF, WISE, PS, Euclid, Roman, SPHEREx, …
- Each one of these is Bn+ objects (w. many more measurements)

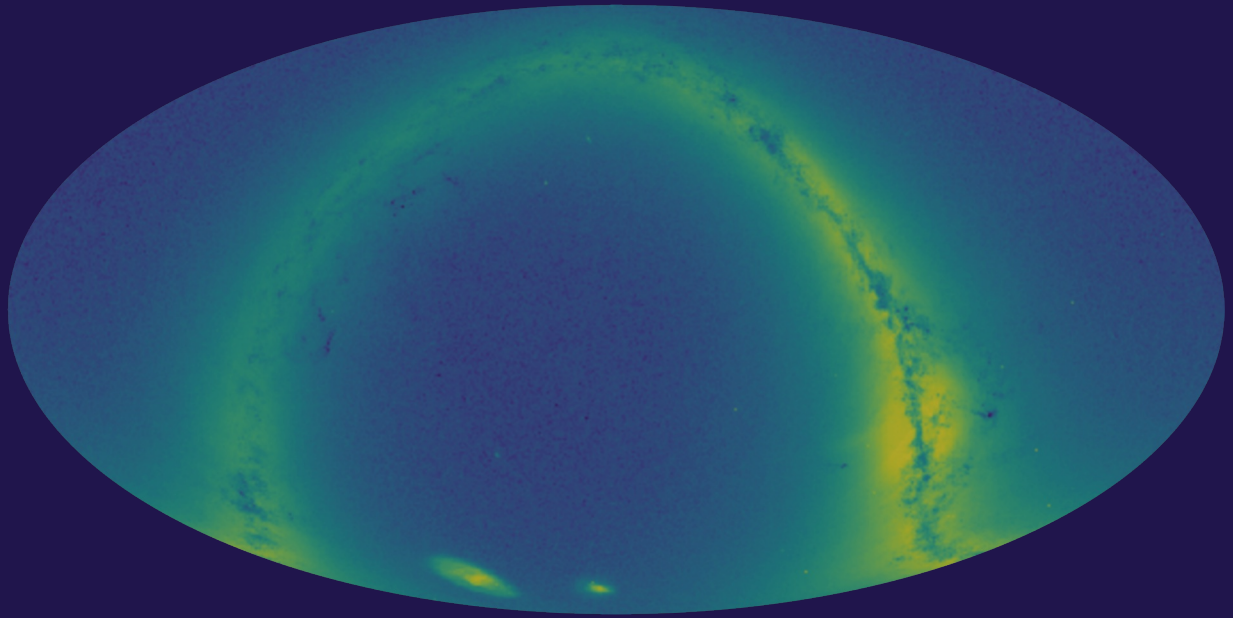
Industry state-of-the-art solution is to use distributed analytics tools (e.g. Spark) on appropriately partitioned files.

Partitioning a large dataset

Gaia DR3

The 1.8Bn sources released in Gaia DR3

A single ASCII file would be about ~680GB in size, (gzip compressed).



1. Partitioning: HEALPix

Partition the sky into NSIDE=1 (order=0) HEALPix tiles, map tiles to files.

Example:

```
Norder0-Npix0.tsv.gz
Norder0-Npix1.tsv.gz
Norder0-Npix2.tsv.gz
Norder0-Npix3.tsv.gz
Norder0-Npix4.tsv.gz
Norder0-Npix5.tsv.gz
Norder0-Npix6.tsv.gz
Norder0-Npix7.tsv.gz
Norder0-Npix8.tsv.gz
Norder0-Npix9.tsv.gz
Norder0-Npix10.tsv.gz
Norder0-Npix11.tsv.gz
```

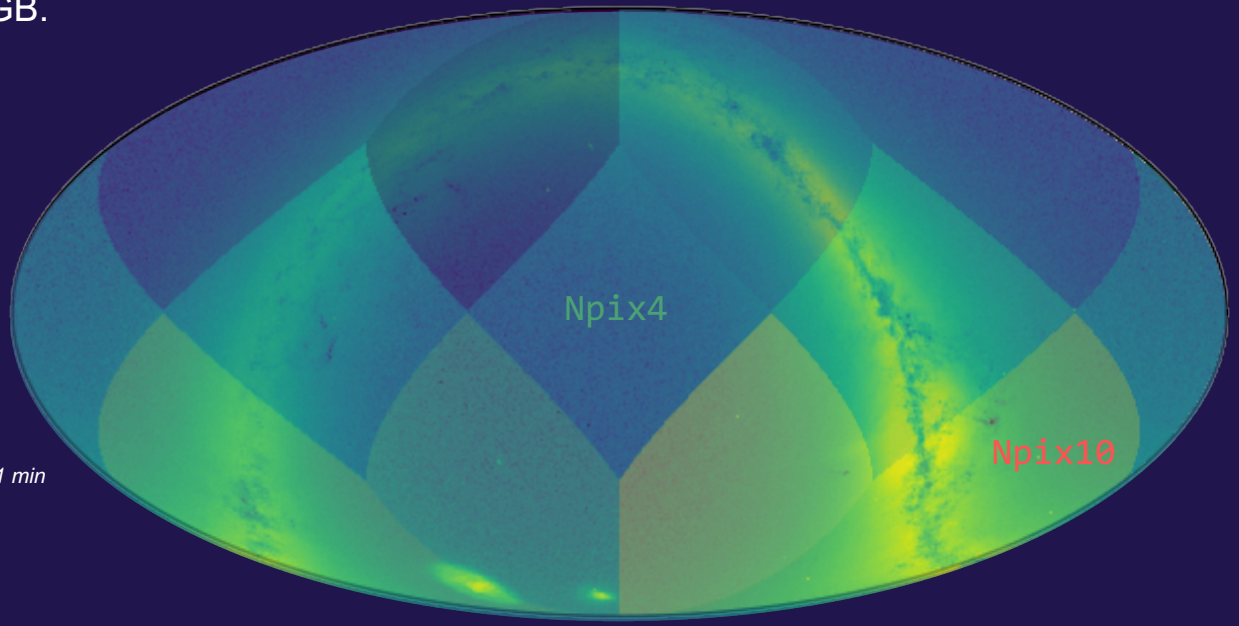
Problem: Severely unbalanced file sizes

Pixel 4 (Galactic pole) ~ 20GB
Pixel 10 (Galactic center) ~ 400GB.

Simple file-based parallelization fails.

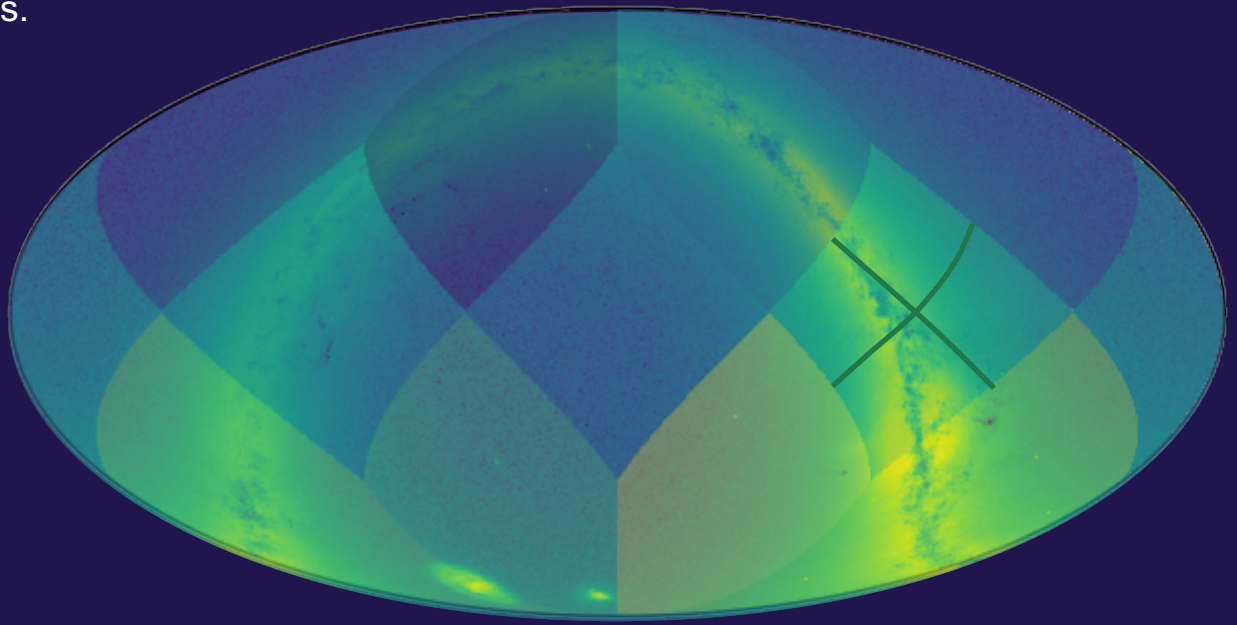
Example

```
Norder0-Npix0.tsv.gz
Norder0-Npix1.tsv.gz
Norder0-Npix2.tsv.gz
Norder0-Npix3.tsv.gz
Norder0-Npix4.tsv.gz   → 1 min
Norder0-Npix5.tsv.gz
Norder0-Npix6.tsv.gz
Norder0-Npix7.tsv.gz
Norder0-Npix8.tsv.gz
Norder0-Npix9.tsv.gz
Norder0-Npix10.tsv.gz  → 20 min
Norder0-Npix11.tsv.gz
```



Solution: Partition Hierarchically

If too many sources fall into a pixel,
split it into four higher order pixels.

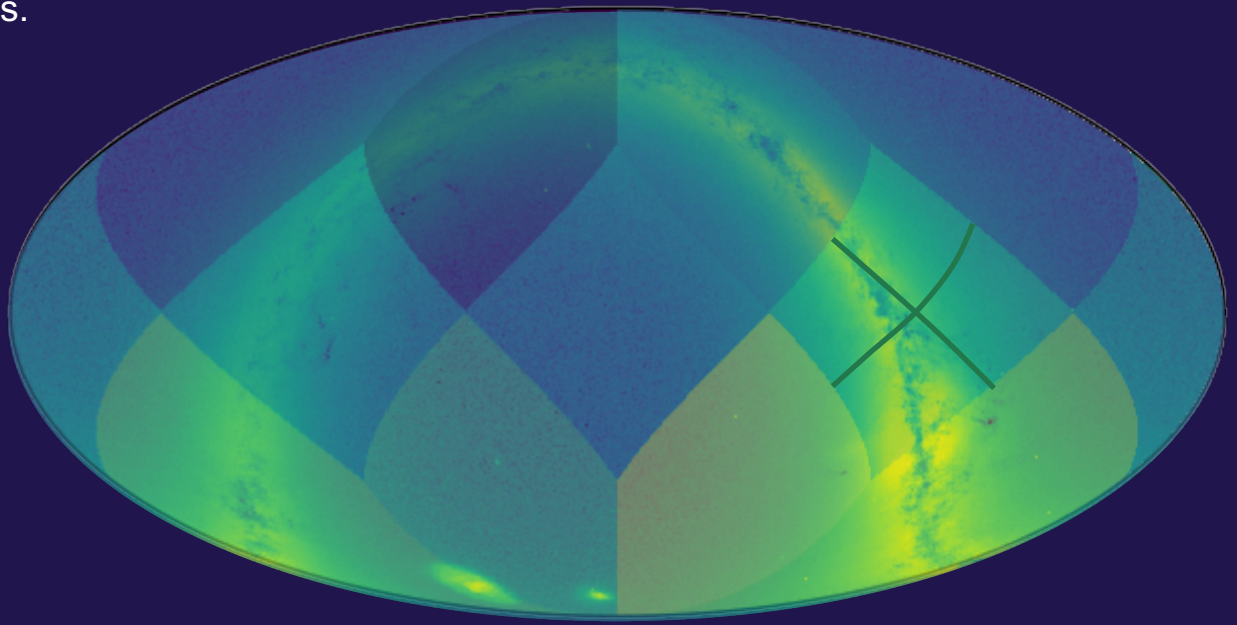


Example

```
Norder0-Npix0.tsv.gz
…
Norder0-Npix7.tsv.gz
…
Norder0-Npix11.tsv.gz
```

1. Partition Hierarchically

If too many sources fall into a pixel,
split it into four higher order pixels.



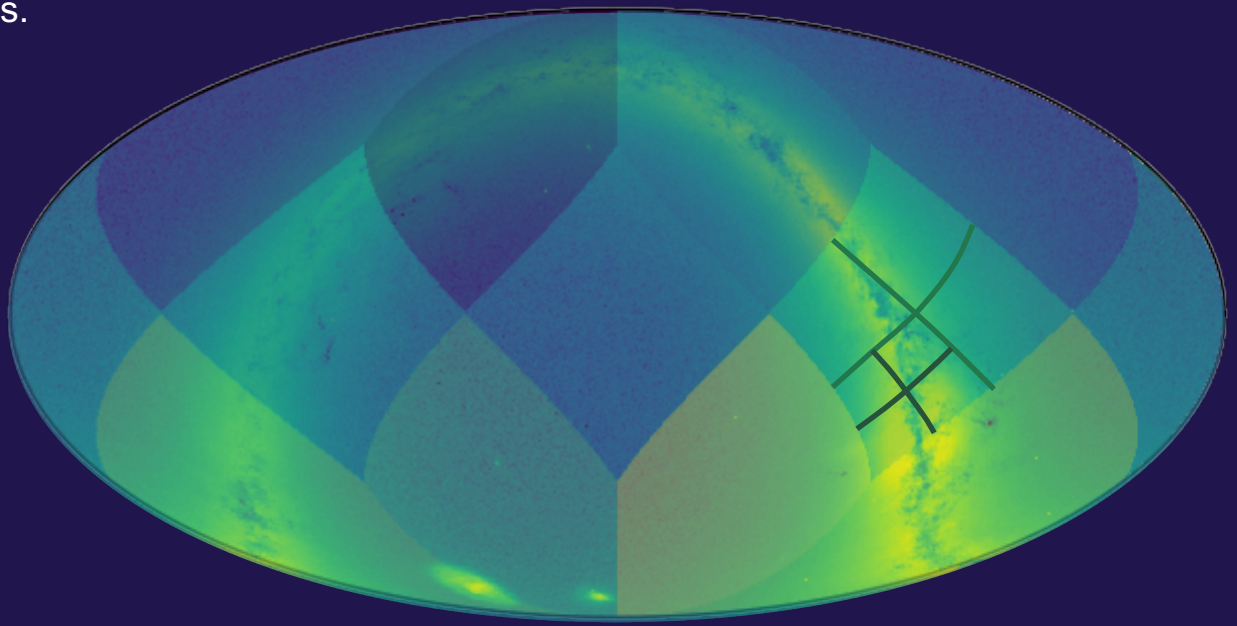
Example

```
Norder0-Npix0.tsv.gz
…
Norder1-Npix28.tsv.gz
Norder1-Npix29.tsv.gz
Norder1-Npix30.tsv.gz
Norder1-Npix31.tsv.gz
…
Norder0-Npix11.tsv.gz
```

1. Partition Hierarchically

If too many sources fall into a pixel,
split it into four higher order pixels.

Repeat.



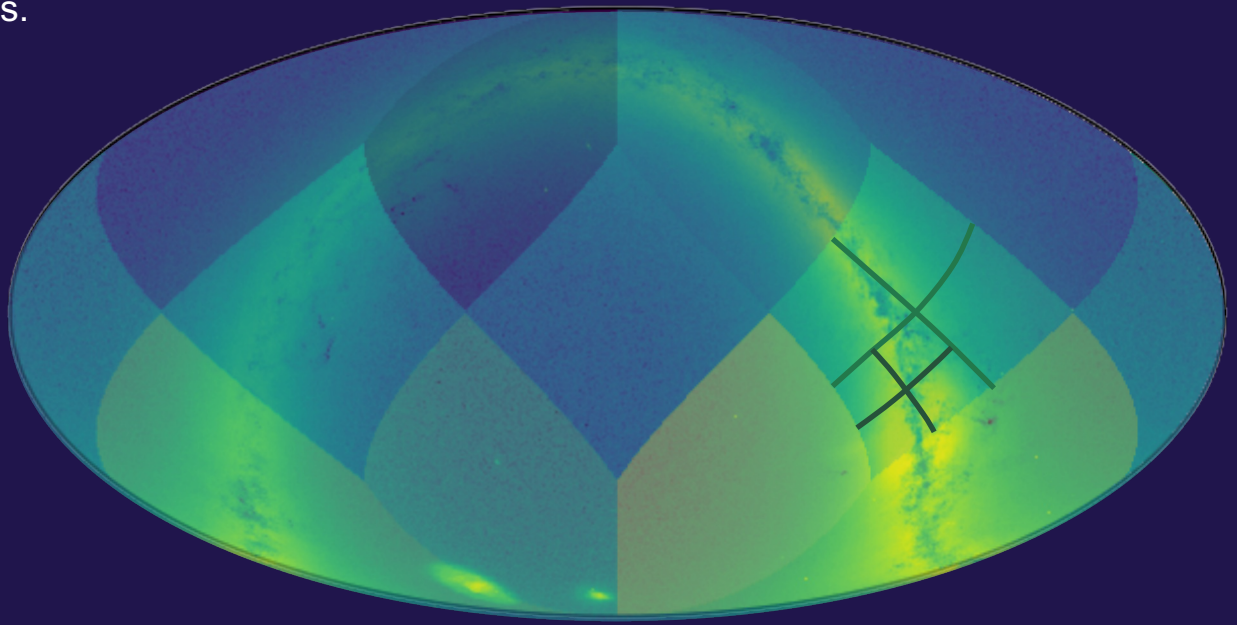
Example

```
Norder0-Npix0.tsv.gz
…
Norder1-Npix28.tsv.gz
Norder1-Npix29.tsv.gz
Norder1-Npix30.tsv.gz
Norder1-Npix31.tsv.gz
…
Norder0-Npix11.tsv.gz
```

1. Partition Hierarchically

If too many sources fall into a pixel,
split it into four higher order pixels.

Repeat.



Example

```
Norder0-Npix0.tsv.gz
…
Norder1-Npix28.tsv.gz
Norder1-Npix29.tsv.gz
Norder1-Npix30.tsv.gz
Norder2-Npix112.tsv.gz
Norder2-Npix113.tsv.gz
Norder2-Npix114.tsv.gz
Norder2-Npix115.tsv.gz
…
Norder0-Npix11.tsv.gz
```

1. Partition Hierarchically

If too many sources fall into a pixel, split it into four higher order pixels.

Repeat until each file size is beneath some pre-defined threshold.

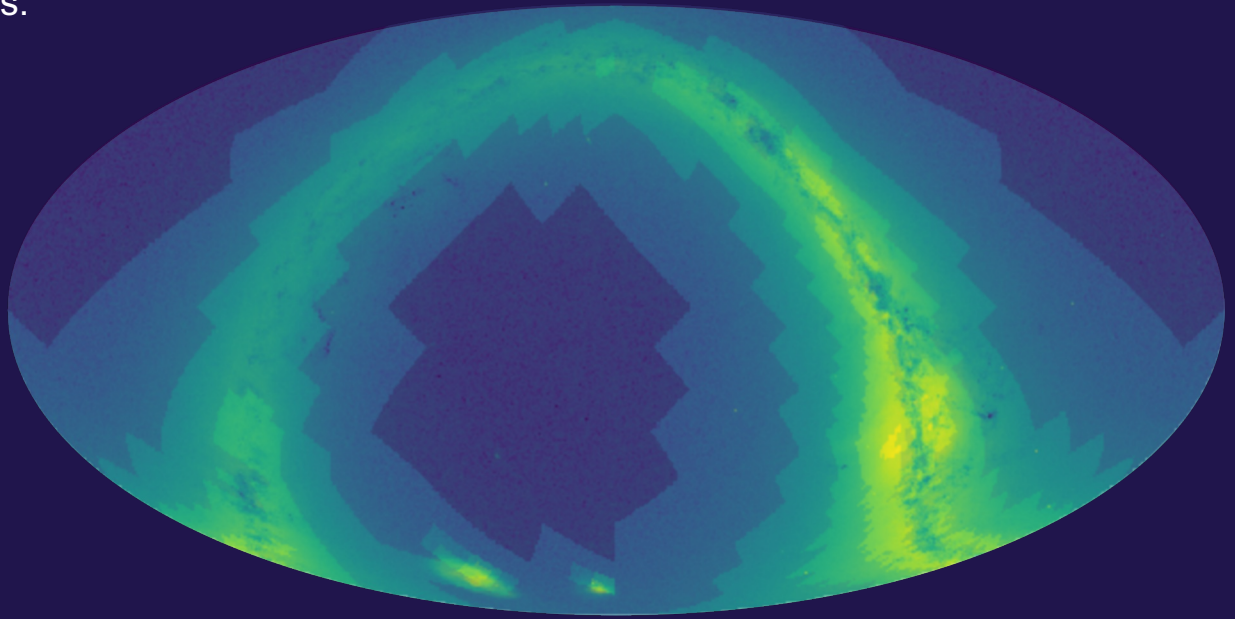


Figure: an overlay of Gaia counts and the partitioning map, taking MAXOBJECTS=1e6

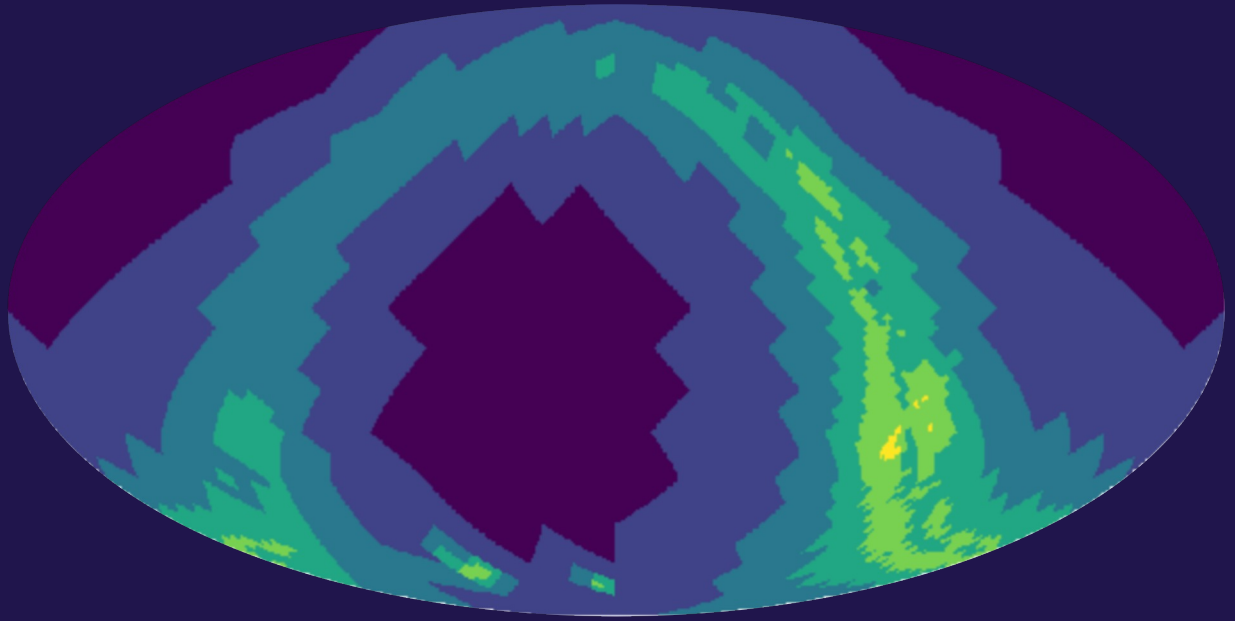
order:

3993 partitions for Gaia DR3, with 1M object/partition threshold

2. On-disk organization

Holding everything in a single directory is unwieldy (at best).

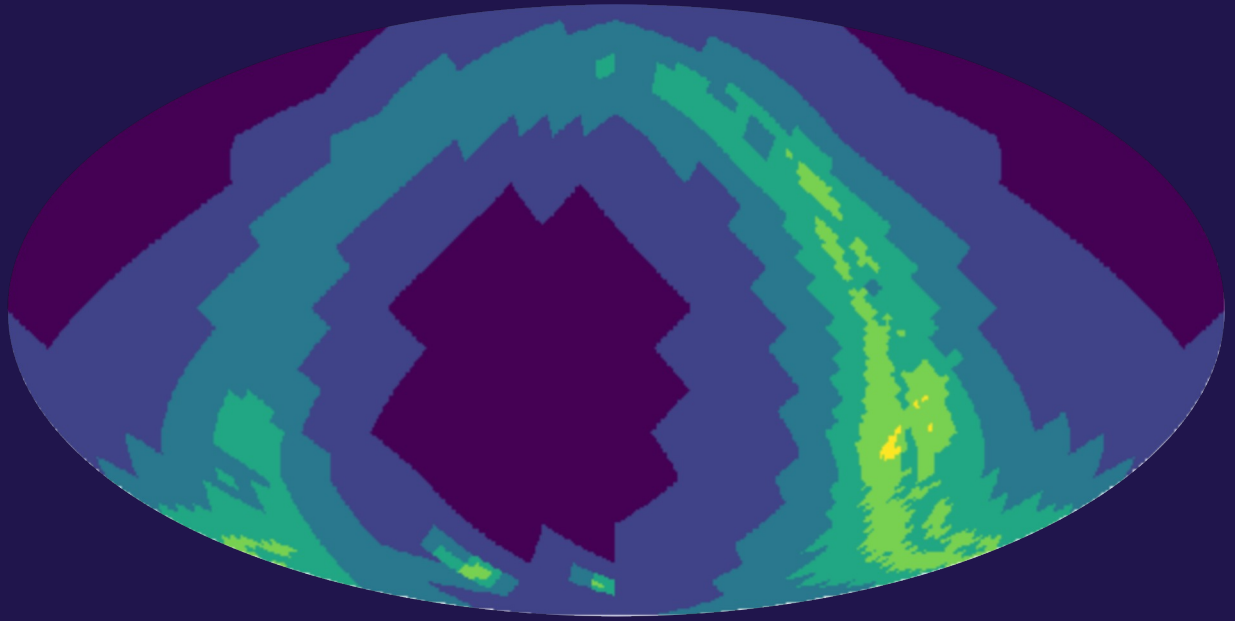
A directory structure encoding the hierarchy would be helpful.



```
Norder0-Npix0.tsv.gz
…
Norder1-Npix28.tsv.gz
Norder1-Npix29.tsv.gz
Norder1-Npix30.tsv.gz
Norder2-Npix112.tsv.gz
Norder2-Npix113.tsv.gz
Norder2-Npix114.tsv.gz
Norder2-Npix115.tsv.gz
…
Norder0-Npix11.tsv.gz
```

2. On-disk organization: HiPS-like Directories

Fortunately, we have a precedent in VO on how to organize hierarchically partitioned HEALPix data: HiPS.



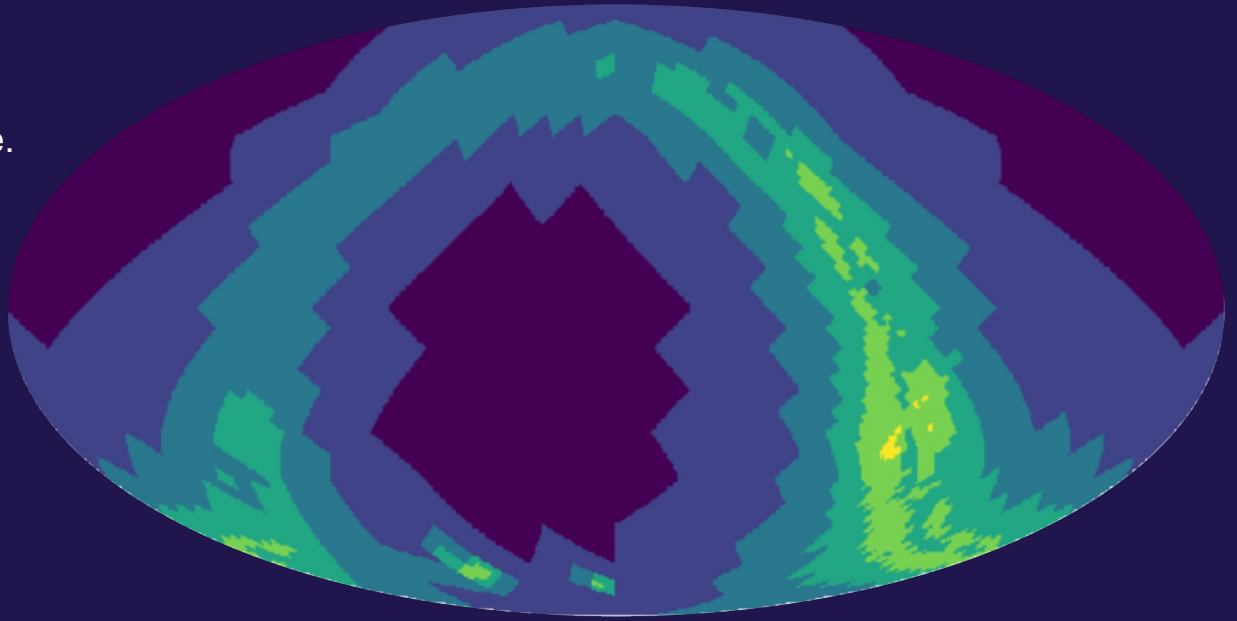
```
Norder0/Dir0/Npix0.tsv.gz
…
Norder1/Dir0/Npix28.tsv.gz
Norder1/Dir0/Npix29.tsv.gz
Norder1/Dir0/Npix30.tsv.gz
Norder2/Dir0/Npix112.tsv.gz
Norder2/Dir0/Npix113.tsv.gz
Norder2/Dir0/Npix114.tsv.gz
Norder2/Dir0/Npix115.tsv.gz
…
Norder0/Dir0/Npix11.tsv.gz
```

3. Serialization

TSV is not ideal for large catalog storage and analytics.

Time-consuming to parse and (de)compress. Also not seekable.

FITS? HDF5?



```
Norder0/Dir0/Npix0.tsv.gz
…
Norder1/Dir0/Npix28.tsv.gz
Norder1/Dir0/Npix29.tsv.gz
Norder1/Dir0/Npix30.tsv.gz
Norder2/Dir0/Npix112.tsv.gz
Norder2/Dir0/Npix113.tsv.gz
Norder2/Dir0/Npix114.tsv.gz
Norder2/Dir0/Npix115.tsv.gz
…
Norder0/Dir0/Npix11.tsv.gz
```

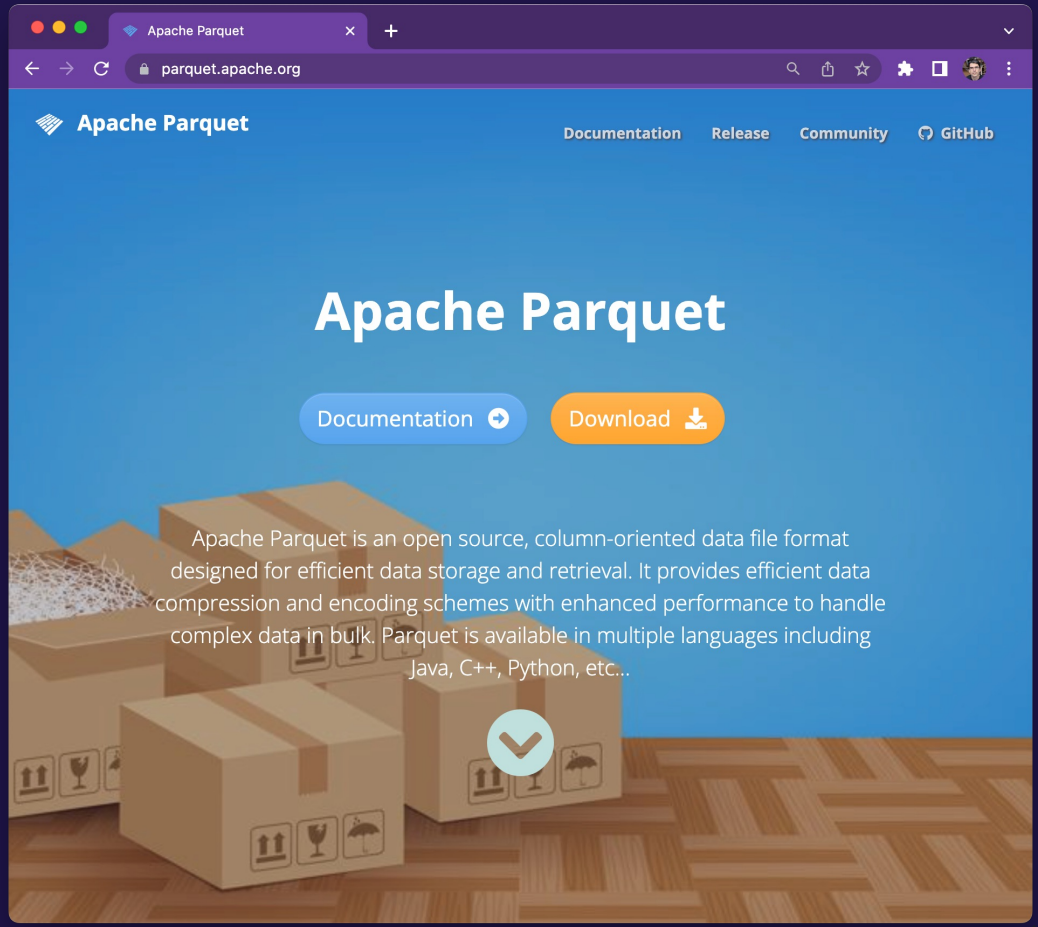
3. Serialization: Parquet

TSV is not ideal for large catalog storage and analytics.

We use Parquet.

<u>Key features:</u>

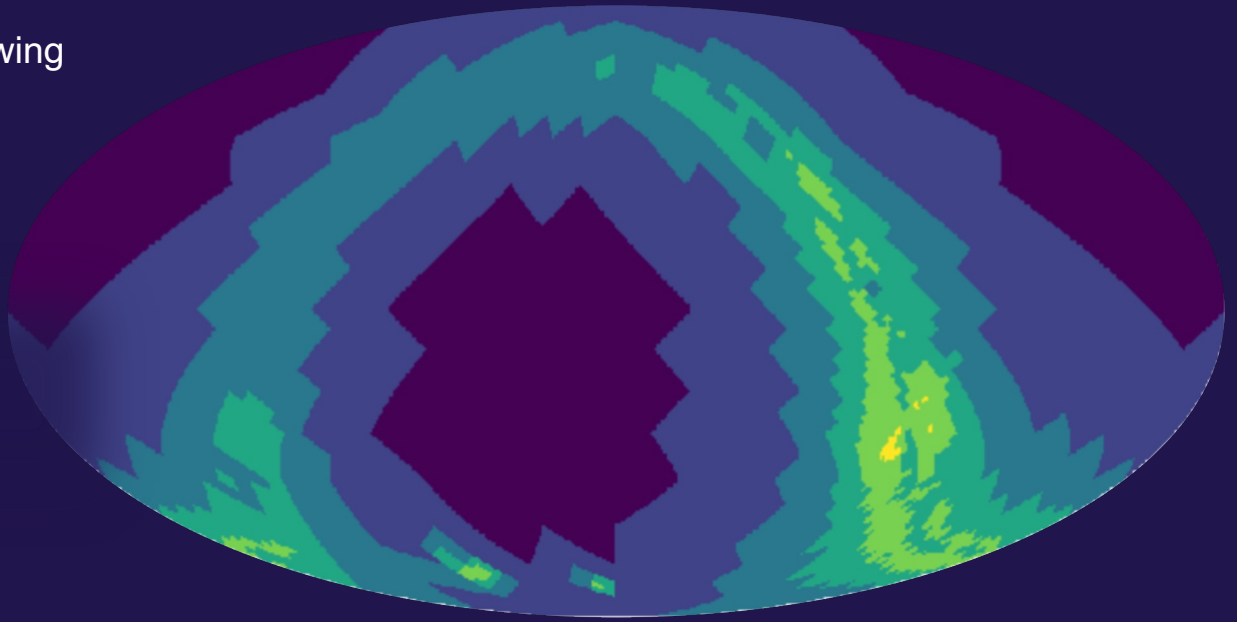
- ✓ Designed for storage of large tables
- ✓ Columnar
- ✓ Efficient (binary)
- ✓ Transparent compression
- ✓ Data Integrity (checksums)
- ✓ Partitioning
- ✓ Broad multi-language support
- ✓ Broad tool support
- ✓ Strong industry backing
- ✓ Open source



3. Serialization: Parquet

Parquet readers natively support reading partitioned datasets if they're stored in directories following `<key>=<value>` naming format.

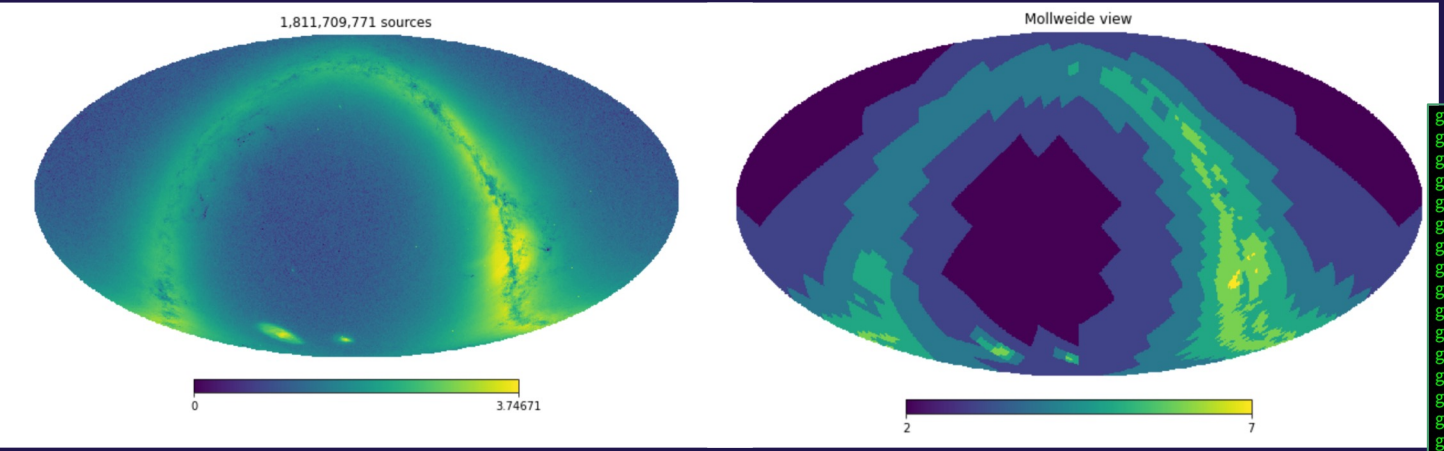
We make that small tweak…



```
Norder=0/Dir=0/Npix=0.parquet
…
Norder=1/Dir=0/Npix=28.parquet
Norder=1/Dir=0/Npix=29.parquet
Norder=1/Dir=0/Npix=30.parquet
Norder=2/Dir=0/Npix=112.parquet
Norder=2/Dir=0/Npix=113.parquet
Norder=2/Dir=0/Npix=114.parquet
Norder=2/Dir=0/Npix=115.parquet
…
Norder=0/Dir=0/Npix=11.parquet
```

(*) Working title. We will change the name to avoid confusion with true HiPS catalogs

All together: HiPSCat*



Gaia DR2 Catalog Counts (log scale)

Visualization of file storage (color = healpix level)
3933 partitions of similar size (128-256 MB)

Layout on "disk": 👇

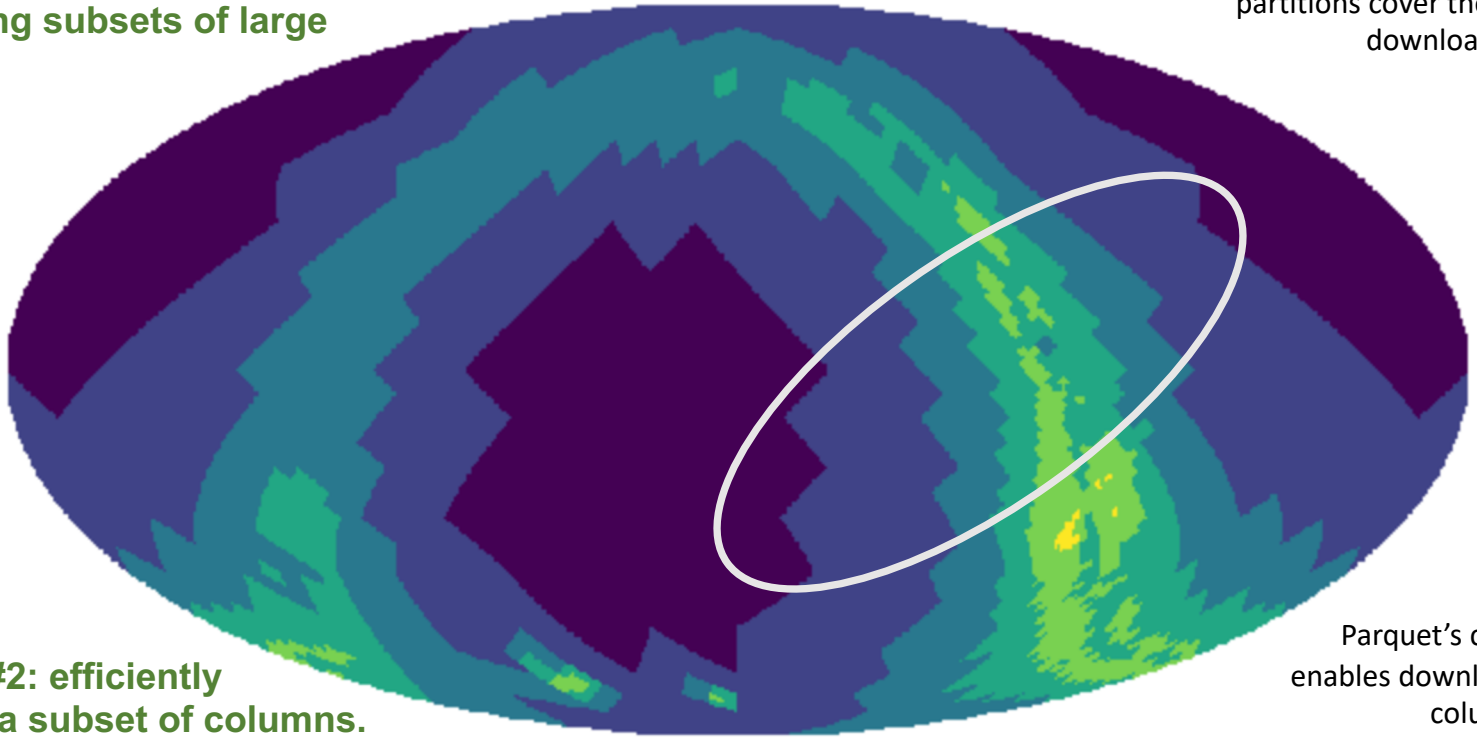
```
gaia/point_map.fits
gaia/_common_metadata
gaia/_metadata
gaia/partition_info.csv
gaia/catalog_info.json
gaia/provenance_info.json
gaia/Norder=7/Dir=110000/Npix=114935.parquet
gaia/Norder=7/Dir=110000/Npix=114892.parquet
gaia/Norder=7/Dir=110000/Npix=117042.parquet
gaia/Norder=7/Dir=110000/Npix=114906.parquet
gaia/Norder=7/Dir=110000/Npix=116742.parquet
gaia/Norder=7/Dir=110000/Npix=116743.parquet
gaia/Norder=7/Dir=110000/Npix=116736.parquet
gaia/Norder=7/Dir=110000/Npix=117017.parquet
gaia/Norder=7/Dir=110000/Npix=117040.parquet
gaia/Norder=7/Dir=110000/Npix=117043.parquet
gaia/Norder=7/Dir=110000/Npix=116738.parquet
gaia/Norder=7/Dir=110000/Npix=114831.parquet
gaia/Norder=7/Dir=110000/Npix=115644.parquet
gaia/Norder=7/Dir=110000/Npix=116739.parquet
gaia/Norder=7/Dir=110000/Npix=116740.parquet
gaia/Norder=7/Dir=110000/Npix=116741.parquet
gaia/Norder=7/Dir=110000/Npix=117041.parquet
gaia/Norder=7/Dir=110000/Npix=117018.parquet
gaia/Norder=7/Dir=110000/Npix=114846.parquet
gaia/Norder=7/Dir=110000/Npix=117019.parquet
gaia/Norder=7/Dir=110000/Npix=115642.parquet
gaia/Norder=7/Dir=110000/Npix=117016.parquet
gaia/Norder=7/Dir=110000/Npix=115645.parquet
gaia/Norder=7/Dir=110000/Npix=115631.parquet
```

What does this let us do?

Download spatial subsets of a catalog

Use case #1: downloading overlapping subsets of large catalogs.

Given a region of the sky, it's straightforward to find which partitions cover the region (and download those files)



Use case #2: efficiently download a subset of columns.

Parquet's columnar layout enables downloading only the columns of interest

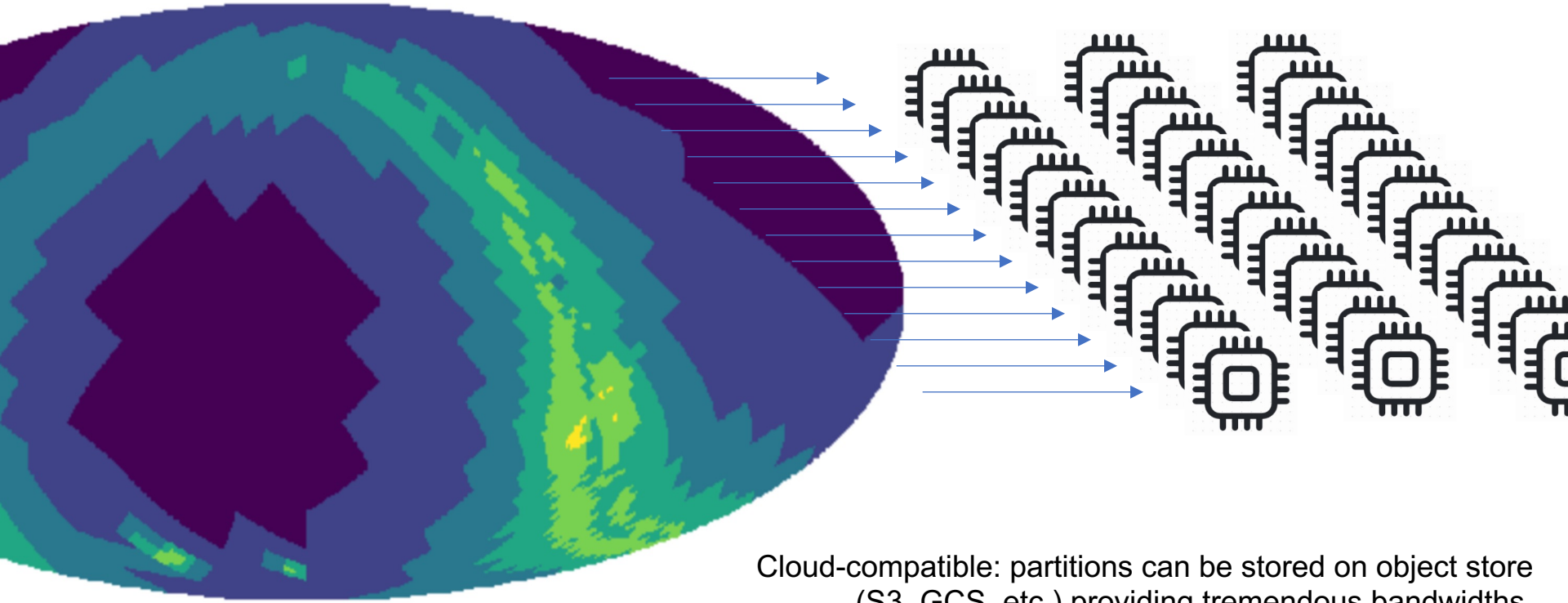
order = 2
pixel size size = 14.7deg

order = 7
pixel size size = 0.46deg

Straightforward Parallel Whole-Catalog Computation

Use case #3: complex searches, feature computation, spatial processing (clustering)

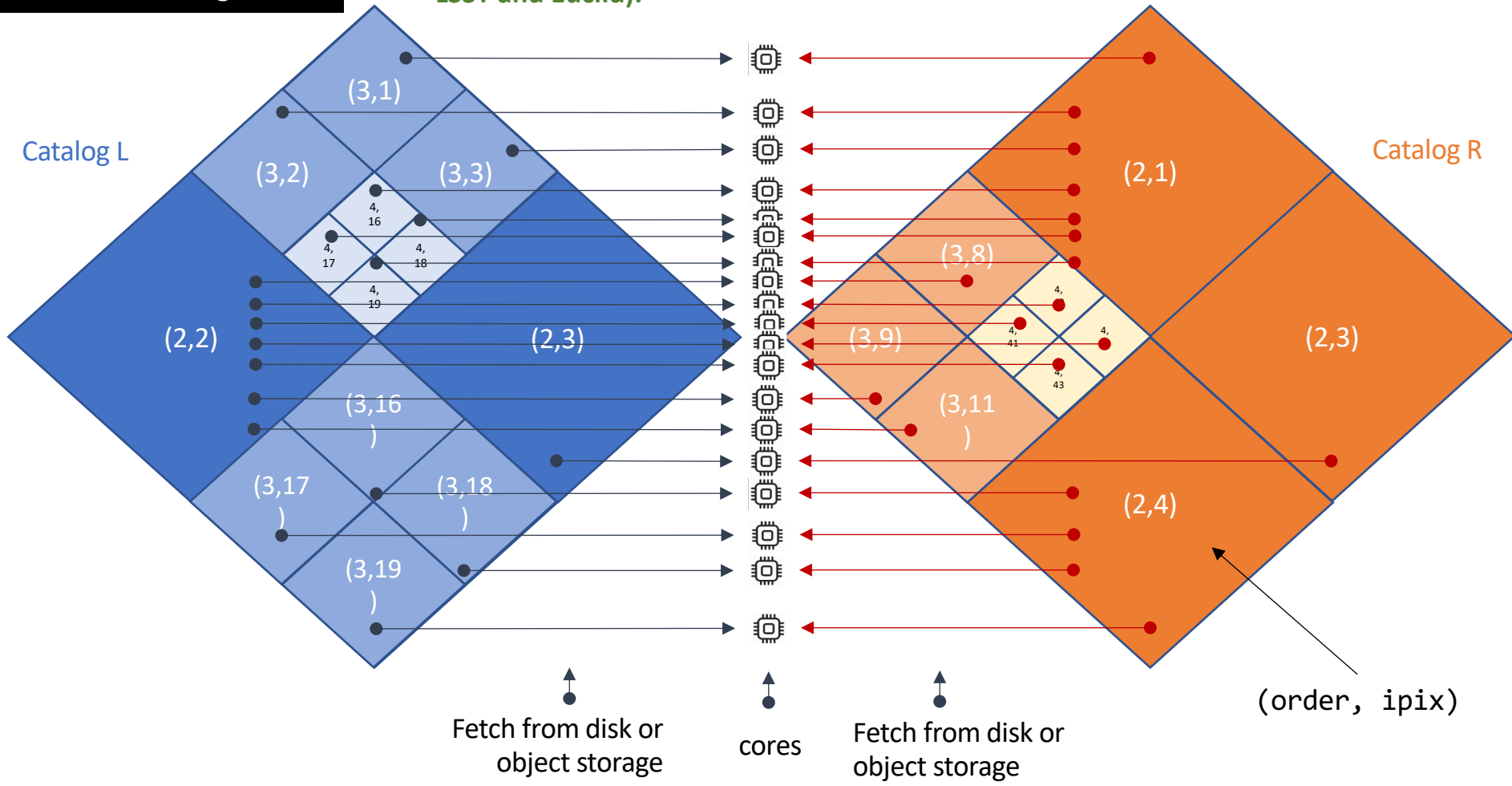
Enables very simple parallel computation schemes: per-file parallelization.



Cloud-compatible: partitions can be stored on object store (S3, GCS, etc.) providing tremendous bandwidths.

Efficient, parallel, joins and crossmatching

Use case #4: distributed analysis on data from two catalogs (example: LSST and Euclid).



Updates since Bologna

Working on this for the past ~1.5yrs

LINCC Team +

Vandana Desai (IPAC), Troy Raen (IPAC), Dave Shupe (IPAC), Brigitta Sipőcz (IPAC)
Gregory Dubois-Felsmann (IPAC & Rubin)
Colin Slater (DiRAC & Rubin)
Sharon Shen (STScI)
Susan Mullally (STScI)
Rick White (STScI)
Bernie Shiao (STScI)
Travis Berger (STScI)
Erik Tollerud (STScI)
Josh Peek (STScI)
Tess Jaffe (HESARC)

(+ YOU, we hope!)

2024 Goals

Providers

- Enabling scalable multi-catalog analytics on large NASA catalog holdings in the NASA Astrophysics Science Platform
- Serving slimmed-down subsets of Rubin data (Brazil LINeA team)

Cloud platforms
Broad user base

IDAC support

Scientists

- Search for unusual variables in ZTF (A. Tzanidakis, UW)
- Search for streams in DELVE+Gaia (Julie Xue, Carnegie Obsv.)
- Probabilistic 3D maps of the Milky Way with Rubin (L. Palaversa, IRB, Croatia)

Timeseries

Join/x-match

Object catalogs

Code is now on GitHub/PyPI/conda-forge

Astronomy Data Commons

Software Infrastructure for Science Platforms and Scalable Astronomy on Cloud Resources

Pinned

Customize pins

axs Public

Astronomy eXtensions for Spark: Fast, Scalable, Analytics of Billion+ row catalogs

Python ☆ 22 11

hipscat Public

Hierarchical Progressive Survey Catalog

Python ☆ 12 2

Low level format routines

lsdb Public

Large Survey DataBase

Python ☆ 9 4

End-user analytics tool

hipscat-import Public

HiPSCat import - generate HiPSCat-partitioned catalogs

Python ☆ 4 2

Robust importer

https://github.com/astronomy-commons

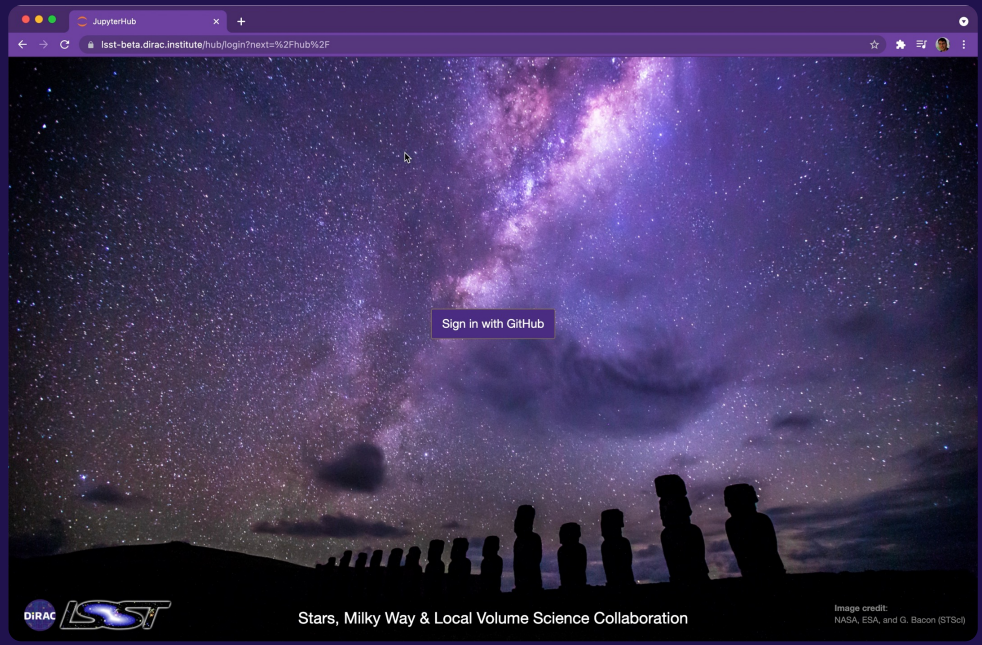
All of these are also available on PyPI and conda-forge.

Test Deployments

LINCC Hub: https://lsst.dirac.dev

- LINCC Hub: *The LINCC Science Platform for the LSST Science Collaborations*
  (email ncaplar@uw.edu for access)

- NASA Astrophysics Science Platform: Working with the Fornax team at IRSA/MAST/HEASARC to deploy and test.



- Or run `conda install -c conda-forge lsdb` on your fav. science platform.

Datasets

- Tested import of a number of existing large catalog datasets, including Gaia, SDSS, DES, PS1, ALLWISE, S-PLUS, Rubin sims, ZTF. No major surprises.
- Learning how to balance number of partitions vs. RAM vs. file size
- Code: https://github.com/astronomy-commons/hipscat-import

```
$ du -kh -d 1 ztf/zubercal/
200M    ztf/zubercal/Norder=2
3.1G    ztf/zubercal/Norder=3
45G     ztf/zubercal/Norder=4
776G    ztf/zubercal/Norder=5
2.1T    ztf/zubercal/Norder=6
3.9T    ztf/zubercal/Norder=7
4.0T    ztf/zubercal/Norder=8
166G    ztf/zubercal/Norder=9
11T     ztf/zubercal/
```

ZTF "ubercal" calibrated catalog (all observations)
600 billion rows spread over 70,853 partitions
median compressed file size: 141M (10Mrow threshold)

```
$ du -kh -d 1 gaia_dr3/gaia
21G     gaia_dr3/gaia/Norder=2
82G     gaia_dr3/gaia/Norder=3
201G    gaia_dr3/gaia/Norder=4
314G    gaia_dr3/gaia/Norder=5
338G    gaia_dr3/gaia/Norder=6
19G     gaia_dr3/gaia/Norder=7
972G    gaia_dr3/gaia
```

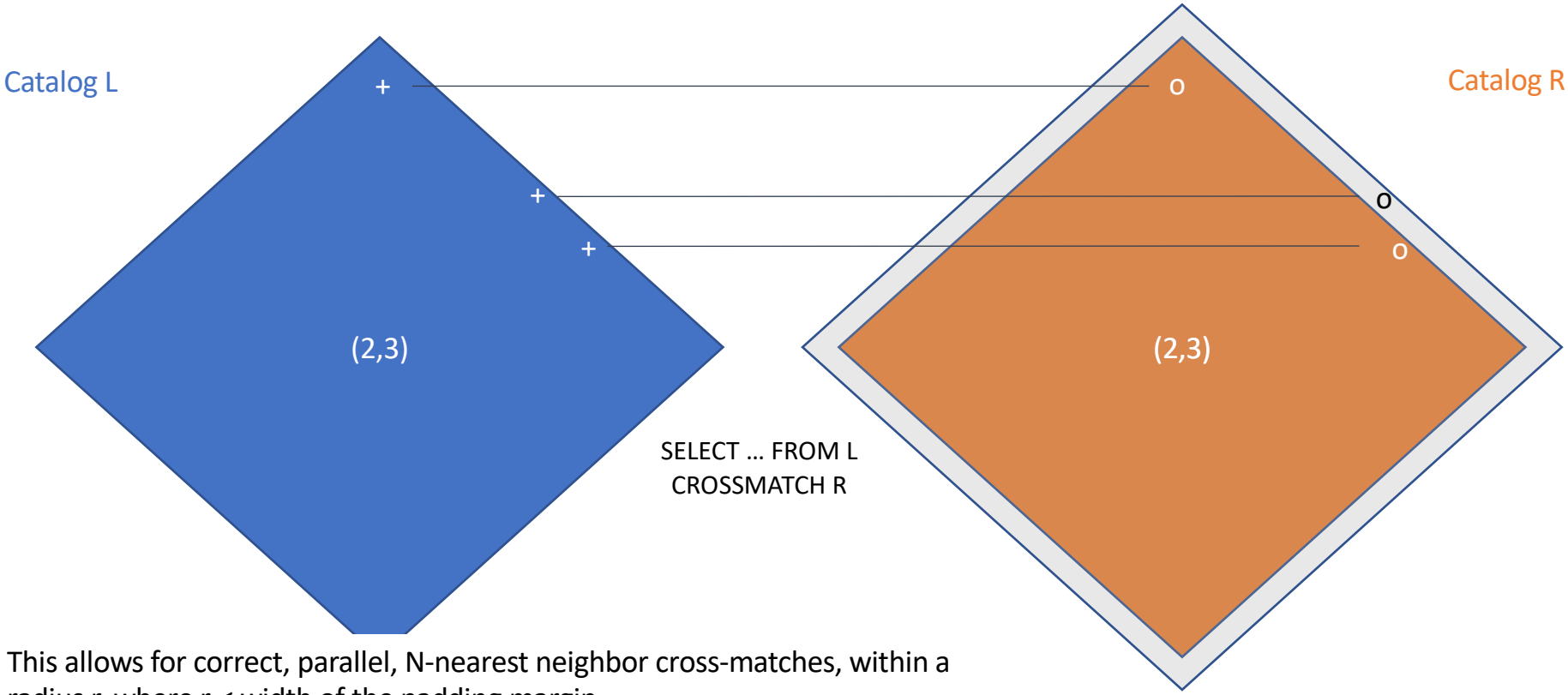
Imported catalogs: https://data.lsdb.io/unstable

Note: we delete/change/replace these on a regular basis. These are /examples/, don't rely on them for serious work!

Distributed cross-match support

Distributed cross-matching and joining

Crossmatching algo: Fetch the coordinate data from both partitions. Make sure to also download the margin of the partition to the right. For each row on the left, find nearest neighbor(s) in the table to the right. In some cases, the NN can be right across the partition boundary, and thanks to the padding, it will be found.



This allows for correct, parallel, N-nearest neighbor cross-matches, within a radius r, where r < width of the padding margin.

Storing the margins: Separate HiPSCat tree

```
$ du gaia_10arcs -kh
20M     gaia_10arcs/Norder=2/Dir=0
20M     gaia_10arcs/Norder=2
162M    gaia_10arcs/Norder=3/Dir=0
162M    gaia_10arcs/Norder=3
719M    gaia_10arcs/Norder=4/Dir=0
719M    gaia_10arcs/Norder=4
1.2G    gaia_10arcs/Norder=5/Dir=0
645M    gaia_10arcs/Norder=5/Dir=10000
1.8G    gaia_10arcs/Norder=5
176M    gaia_10arcs/Norder=6/Dir=10000
1.4G    gaia_10arcs/Norder=6/Dir=20000
507M    gaia_10arcs/Norder=6/Dir=30000
1.5G    gaia_10arcs/Norder=6/Dir=40000
3.5G    gaia_10arcs/Norder=6
402M    gaia_10arcs/Norder=7/Dir=110000
402M    gaia_10arcs/Norder=7
6.6G    gaia_10arcs
```

We're storing the margin data into a separate HiPSCat tree.

This keeps the original catalog "clean", allows for existance of multiple margins, as well as third-party generation of margins.

<u>Key Known Issue:</u>

- At present, it's a 1:1 map in terms of partitioning. This generates numerous small files (e.g. 2M file median, for Gaia).
- Planning to allow independent partitioning of margin data vs. main catalog data.

Tool Support

LSDB: Python Analytics for HiPSCat

- LSDB: Large Survey Database
- Enable Pandas-like analysis on trillions of observations with thousands of cores
- Build on existing tools: Dask (looking at Ray).
- Full HiPSCat awareness: spatial queries, cross-matching, timeseries, multi-dataset joining.
- Alpha-quality (in particular, the API is far from stable).

```
img = gaia
      .query("pm > 10")
      .crossmatch(ztf)
      .join(ztf_sources)
      .for_each(varstar_classify)
      .query("pRRLy > 0.95")
      .skymap()

hp.mollview(img)
```

LSDB target APIs: The API center science. Multi-processing, autoscaling, fail-over, etc. are all implicit. Good user experience.

Wyatt et al. (2023)
https://github.com/astronomy-commons/lsdb

Brief Demo

(You can run this yourself – clone from https://github.com/lincc-frameworks/IVOA_2024_demo)

Summary and next steps

Mailing list: https://groups.google.com/g/hipscat-wg
Repositories: https://github.com/astronomy-commons
Meetings: *10am PT, every third Friday of the month*

- Aiming to enable end-user analyses on 1-100T+ catalog datasets. Ad-hoc collaboration of scientists/engineers from LINCC, Rubin, MAST, IRSA, HESARC, LINeA (Brazil). Developing formats and tools.

- Lots of progress since last year. Data format and Python tools solidifying. We've gathered enough initial real-world usage to start drafting the formal format spec. How do we engage effectively w. the IVOA? (our core engineers are not yet in IVOA WGs).

- Next year:
  - Perfect the tools (UX and performance are a priority)
  - Larger scale user deployment
  - Would love to get addt'l feedback and implementations (e.g. Java or Rust)


LINCC
Carnegie Mellon University
UNIVERSITY OF WASHINGTON
LVX SIT
1861
Collaboratively advancing data-intensive astronomy.
LSST Discovery Alliance
NSF
SCHMIDT FUTURES
AST-2003196
Thank You !
Contact: mjuric@uw.edu
UNIVERSITY of WASHINGTON