EXTREME SCALE SURVEY SIMULATION WITH PYTHON WORKFLOWS

Sierra Villarreal, Postdoctoral Appointee,
Argonne National Laboratory
Oct 28, 2021, ParslFest

Extreme Scale Survey Simulation with Python Workflows

LSST DESC DC2 Image Simulation Campaign

imSim Workflow w/ Parsl Overview

Experiences of Production

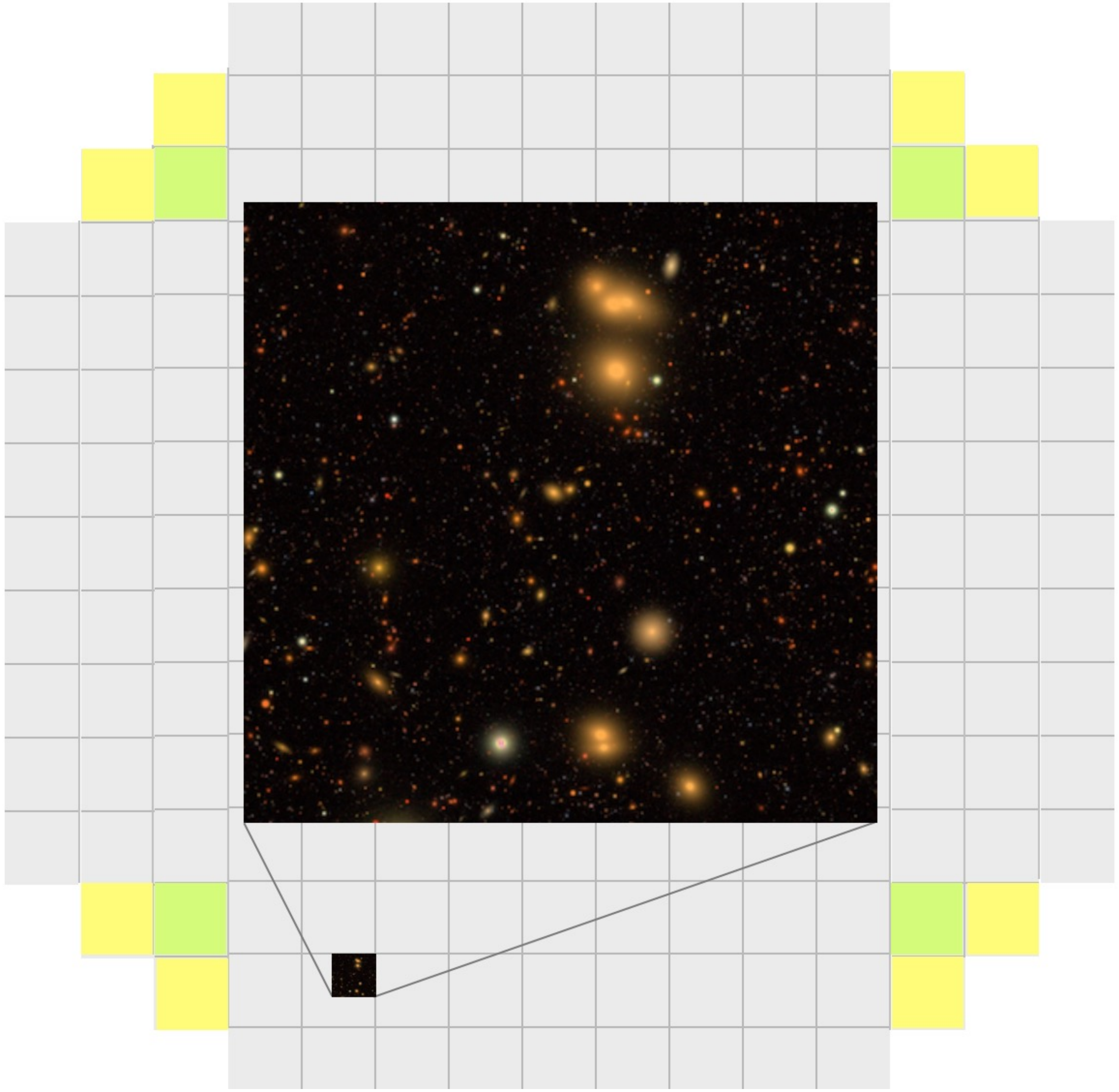
Lessons Learned

LSST DESC DC2 Image Simulation Campaign

- Vera Rubin Observatory LSST is an extreme scientific endeavor in astronomy
  - Large amount of data for the field
  - Significant need to minimize systematic error for science
  - Scientific pipelines need extensive testing
- LSST Dark Energy Science Collaboration (DESC) running Data Challenges!
  - Allows us to build and test pipelines on realistic data
  - DC2 campaign is the most elaborate effort on this front
    - Simulated telescope images
    - 300 square degrees, 5 years deep
    - Multiple color bands
- Two Component Simulation
  - Wide-Fast-Deep (WFD) survey for large area
  - Deep Drilling Field (DDF) survey for transient objects studies (small area)

DC2 Image Simulation

- DC2 consists of over 30,000 telescope observations called “visits”
- Each visit consists of 189 sensors, depicted in gray
- Each sensor shares information about a shared sky for the visit, but must simulate entirely individual objects
- Some visits chosen to contain transient objects for science verification



imSim Workflow with Parsl

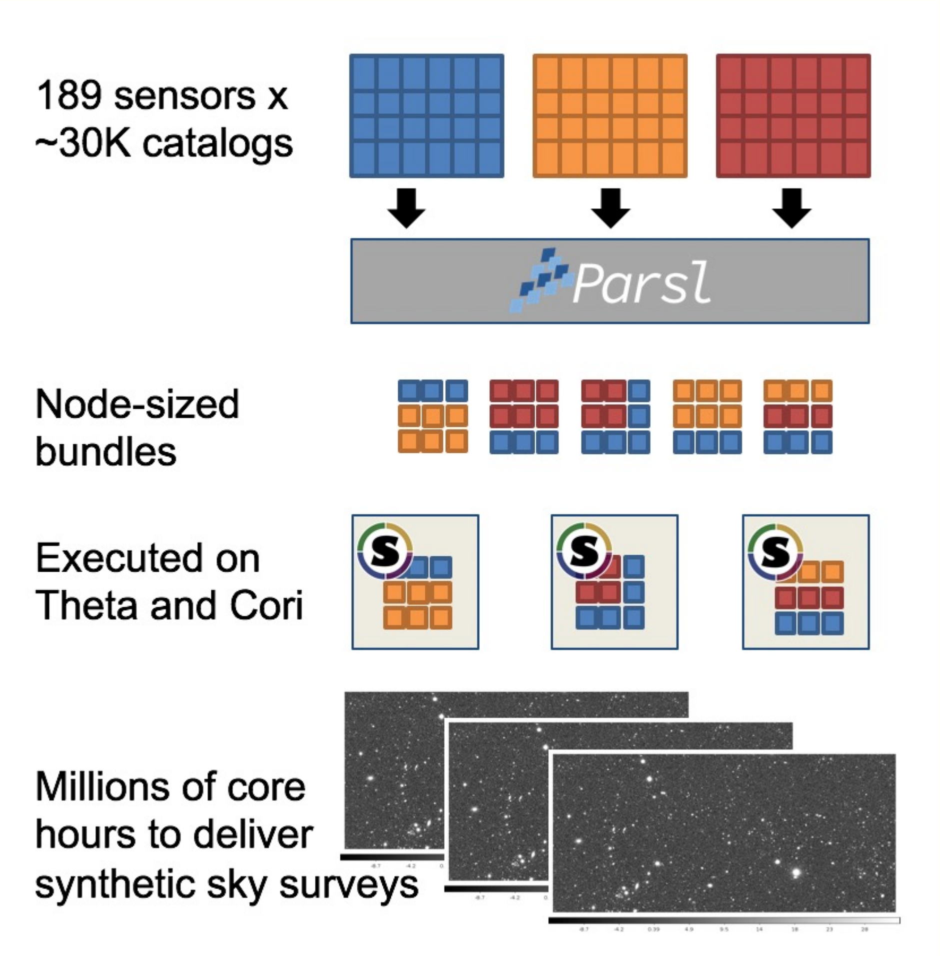
imSim

- Inputs:
  - Catalog of objects to draw for an observation + information about the telescope
  - List of sensors to simulate
  - DDF uses input checkpoint files
- Work:
  - Simulates intervening sky
  - Determines which objects influence a sensor
  - Simulates those objects on the camera
- Outputs:
  - Raw telescope quality image

Parsl

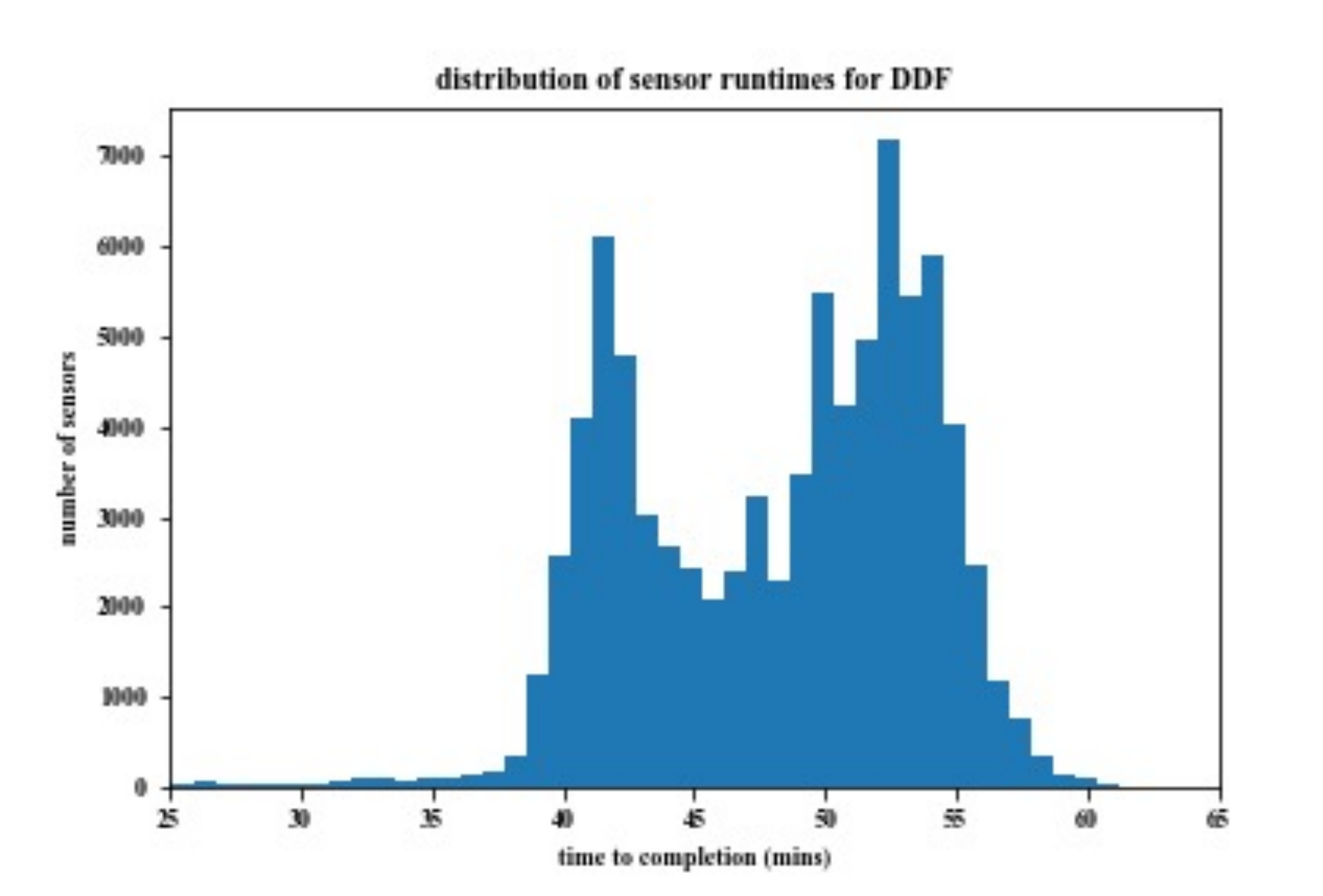
- Parsl driver running on compute resources on allocation
- Runs pre-processing Python scripts to package imSim tasks into bundles
- Runs executors on each compute note on allocated reosurce
- Workers start "Docker" container
- Workers run imSim tasks inside container

Workflow Diagram



Sensor Time Distribution

- Runtimes to Sensor Image for Y3 DDF
- Note — large time discrepancy
  - Time discrepancy *not* just visit to visit
  - Individual objects appearing on one sensor can have long run times
  - Incredibly hard to predict
- Worse for WFD
  - Runtimes on O(ten hours), with ~10% sensor to sensor changes



Production Highlights

~2000 nodes simultaneously utilized on Cori KNL resource at NERSC

~2800 nodes simultaneously utilized on Theta KNL resource at ALCF

Workflow flexibly transferred between these two resources

~4000 nodes utilized on Theta KNL during prototyping (scaled own due to queueing policy quirks)

~100M Compute Hours used across both sites

Lessons Learned

- Parsl processes can disconnect from compute side workers
  - Remotely Driven Workflows (AWS? funcx?)
  - Utilization of Workflow Nodes (NERSC and ALCF both exploring)
- Further Containerization Studies
  - Do we gain or lose time by using containers?
  - If we lose time, do we lose enough that we need to worry?
- Code Improvements Toward Parallelization
  - Downside of containers is that we can’t put new work **into** the container in this workflow
  - Can either minimize lost time in the container or work around this otherwise

Conclusions

- Largest Survey Simulation
- Workflow Entirely Python Driven
- Flexible Containerization
- Workflow Itself Flexible
  - Parsl image processing
  - Parsl science pipeline

Vera Rubin Observatory

THANKS FOR LISTENING!

avillarreal@anl.gov works for any extended questions!