Gravitational Waves - LIGO-Virgo-KAGRA collaboration

Gareth Cabourn Davies [gareth.cabourn-davies@port.ac.uk], University of Portsmouth
and Daniel Williams [daniel.williams@glasgow.ac.uk], University of Glasgow / GEO
GridPP 52, 30 August 2024

Slides adapted from a talk by D. MacLeod, V. Raymond, S. Fairhurst

LVK Science

Origins of Gravitational Waves

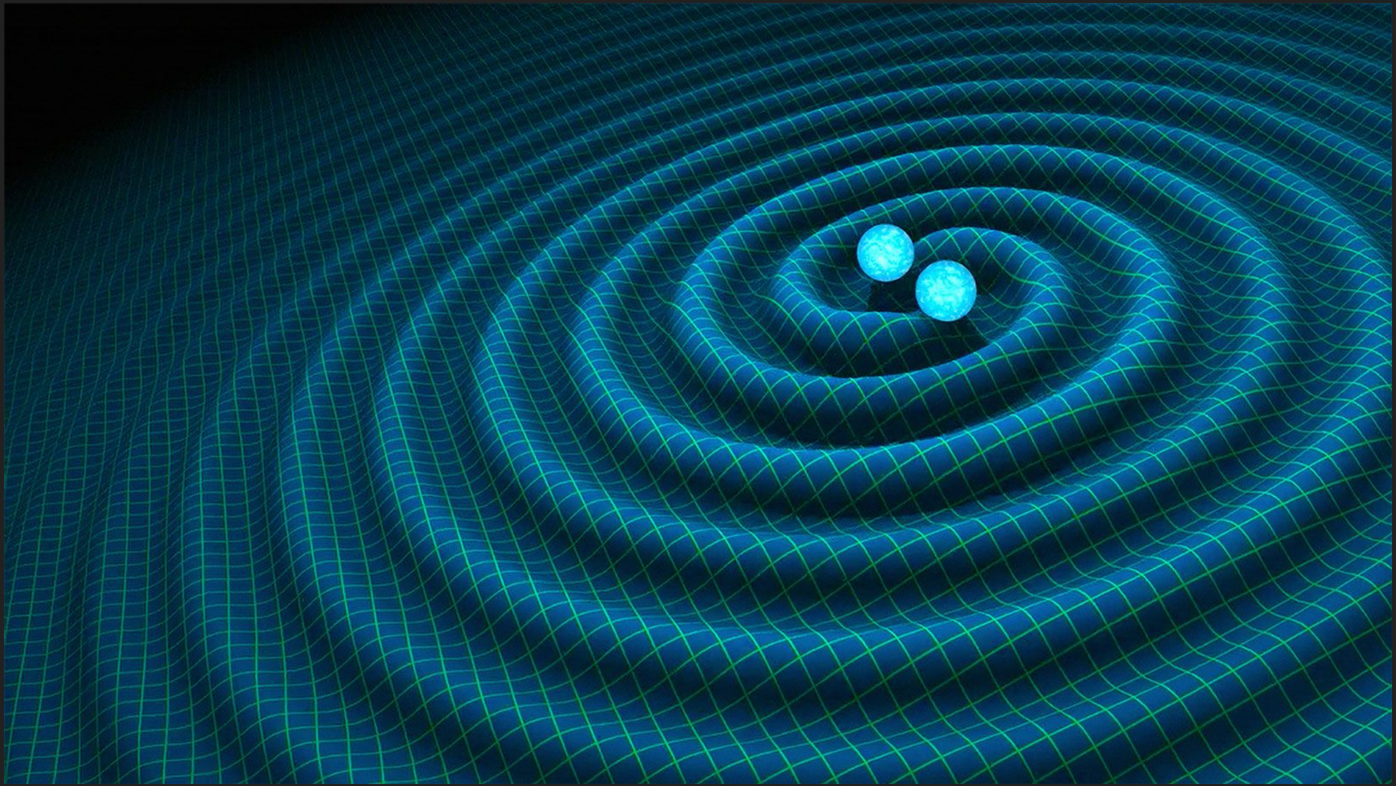
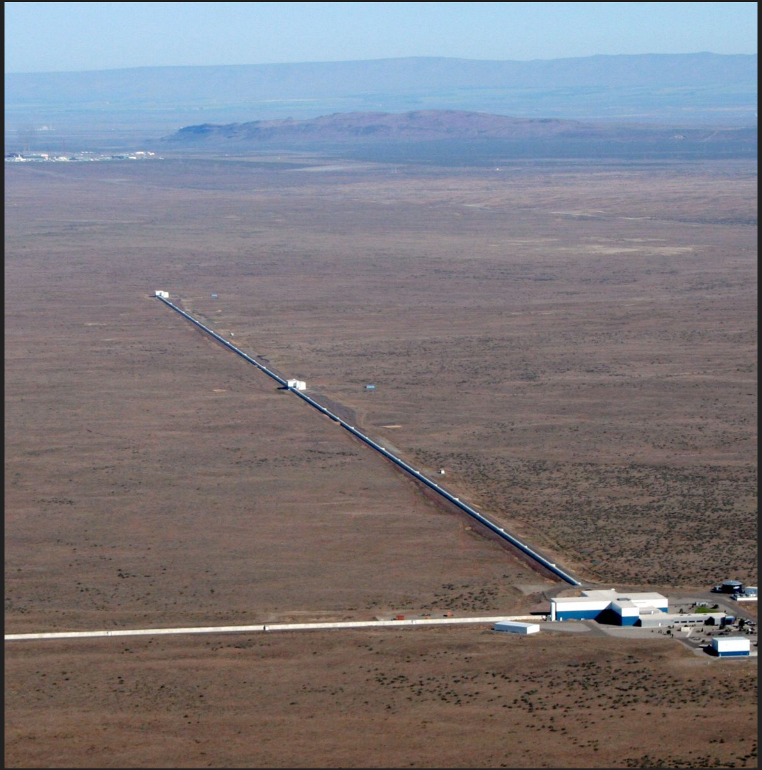
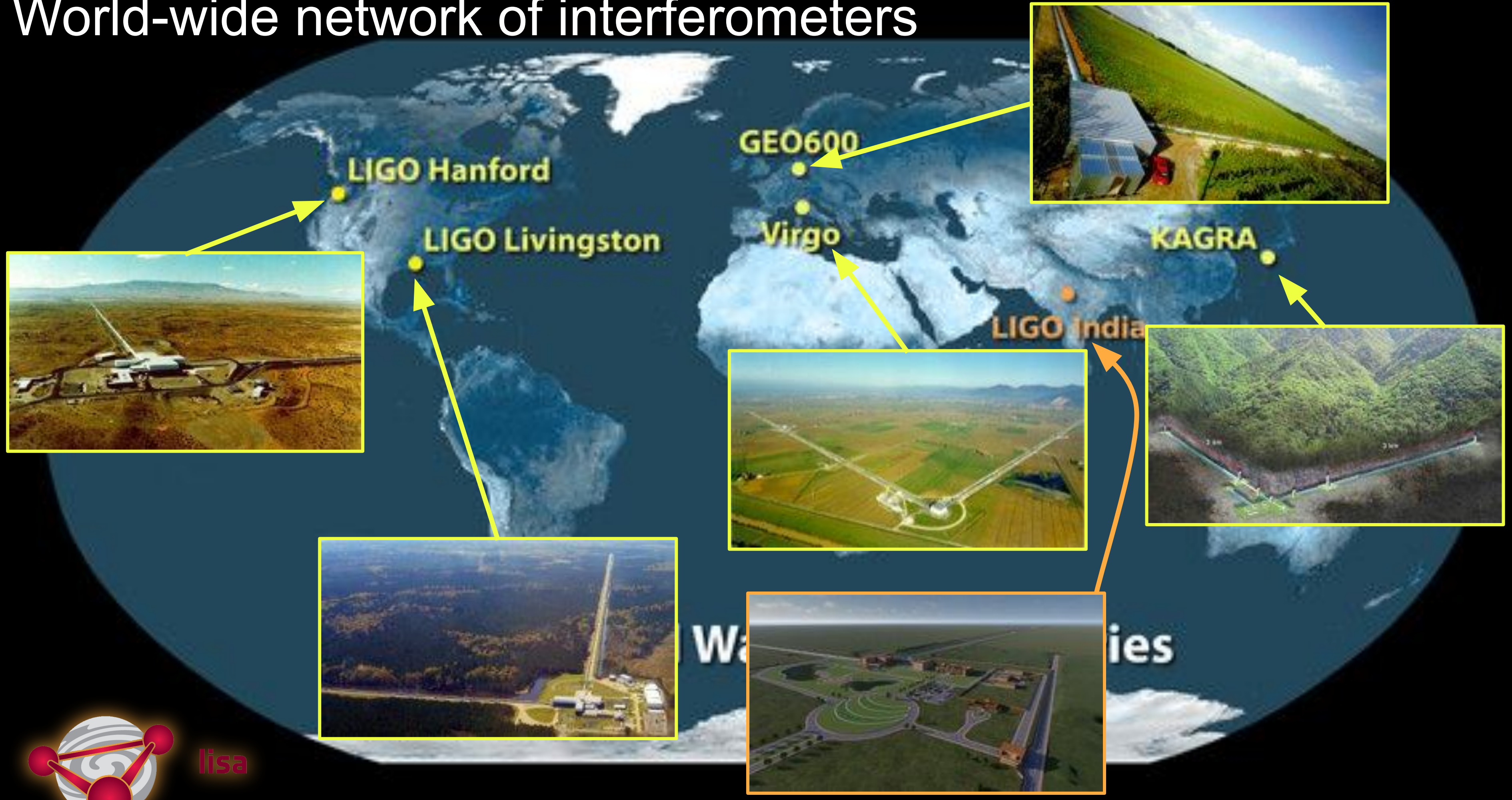


Image: R Hurt / Caltech-JPL

Interferometers



World-wide network of interferometers



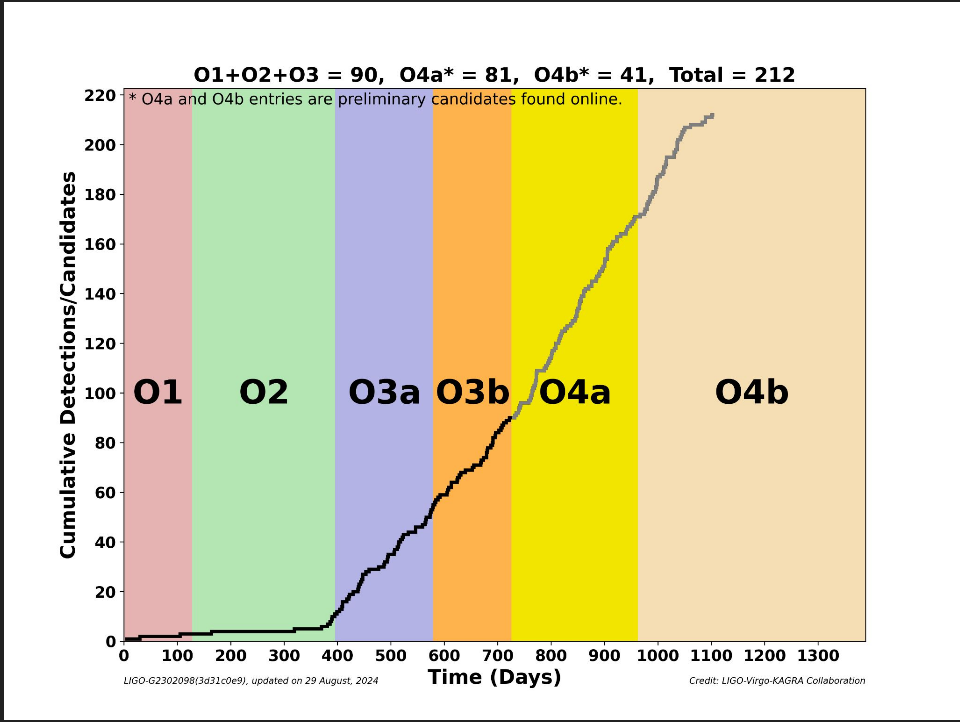
lisa

Observing Runs
Finding Events

Searching through the data for events that looks like the signatures we expect to find

- **O1** (09/2015 - 01/2016)
- **O2** (11/2016 - 08/2017)
- **O3a** (04/2019 - 20/2019)
- **O3b** (11/2019 - 03/2020)
- **O4a** (05/2023 - 01/2024)
- **O4b** (04/2024 - 06/2025)

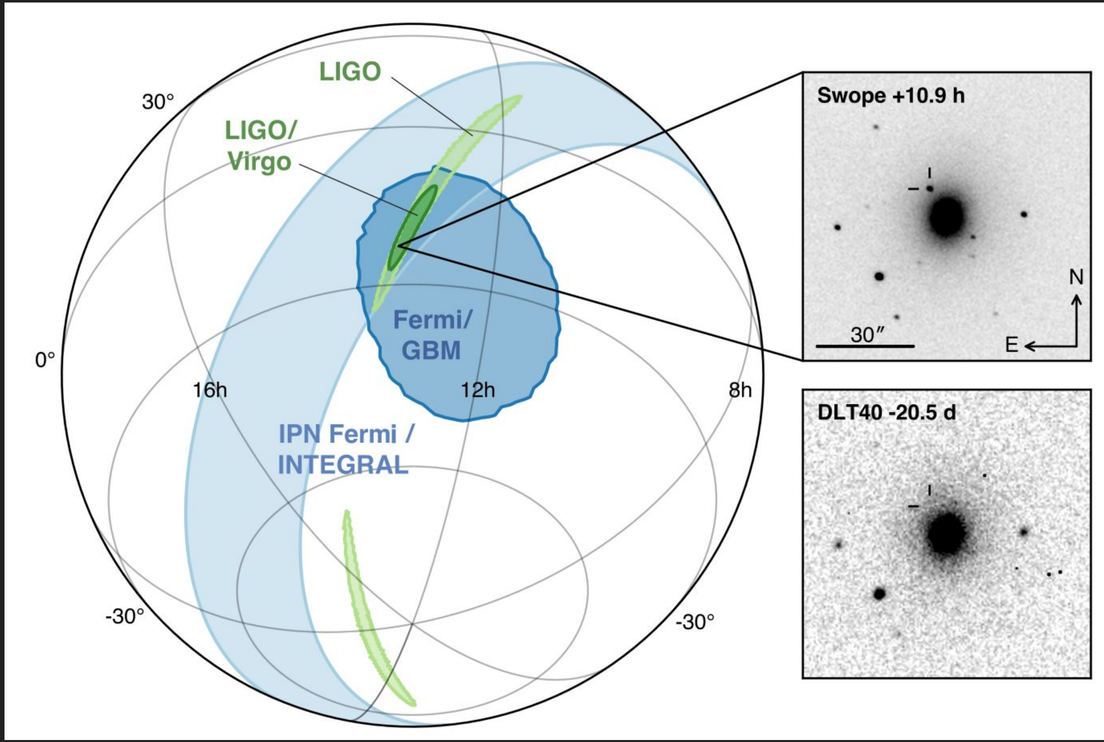
Event rate scales as the sensitive range cubed



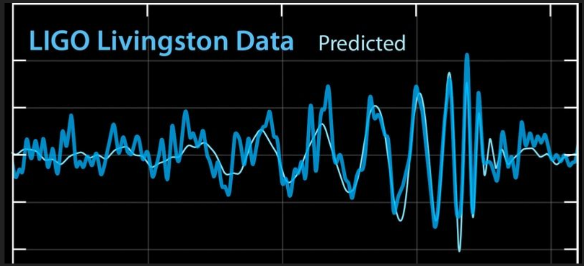
Low Latency - multimessenger signals

Rapid analysis allows us to see gravitational waves and electromagnetic counterparts in real time

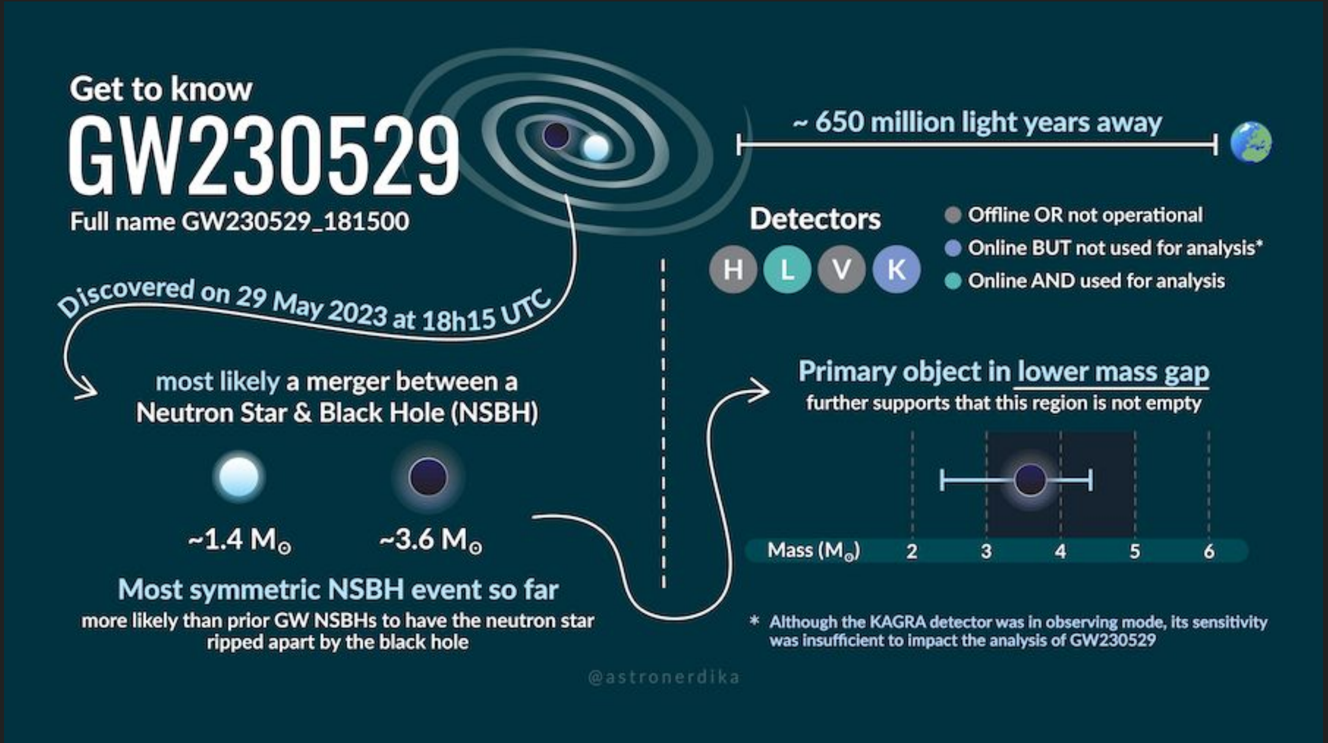
This happened for GW170817



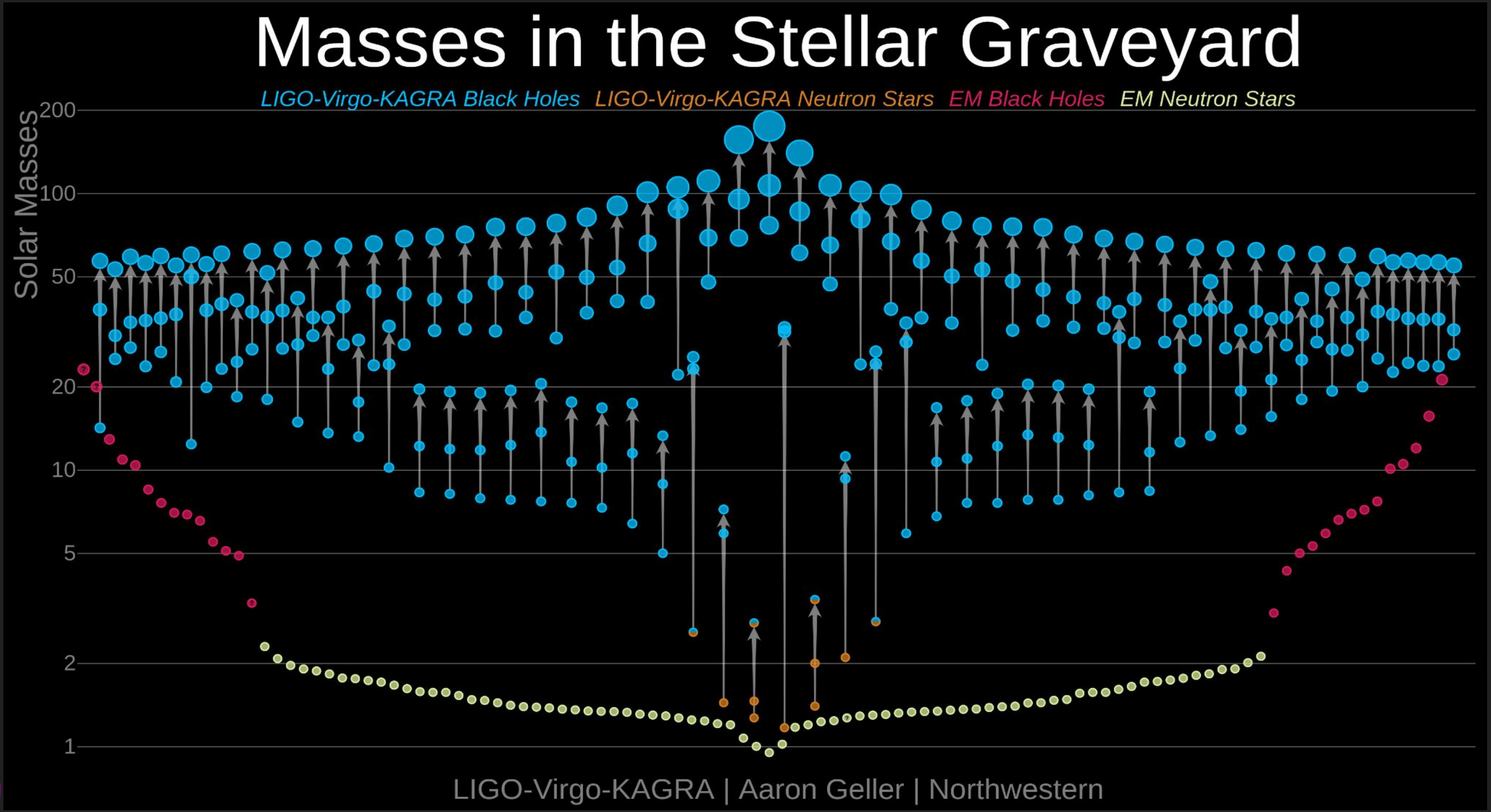
Observing Runs
Categorising Events



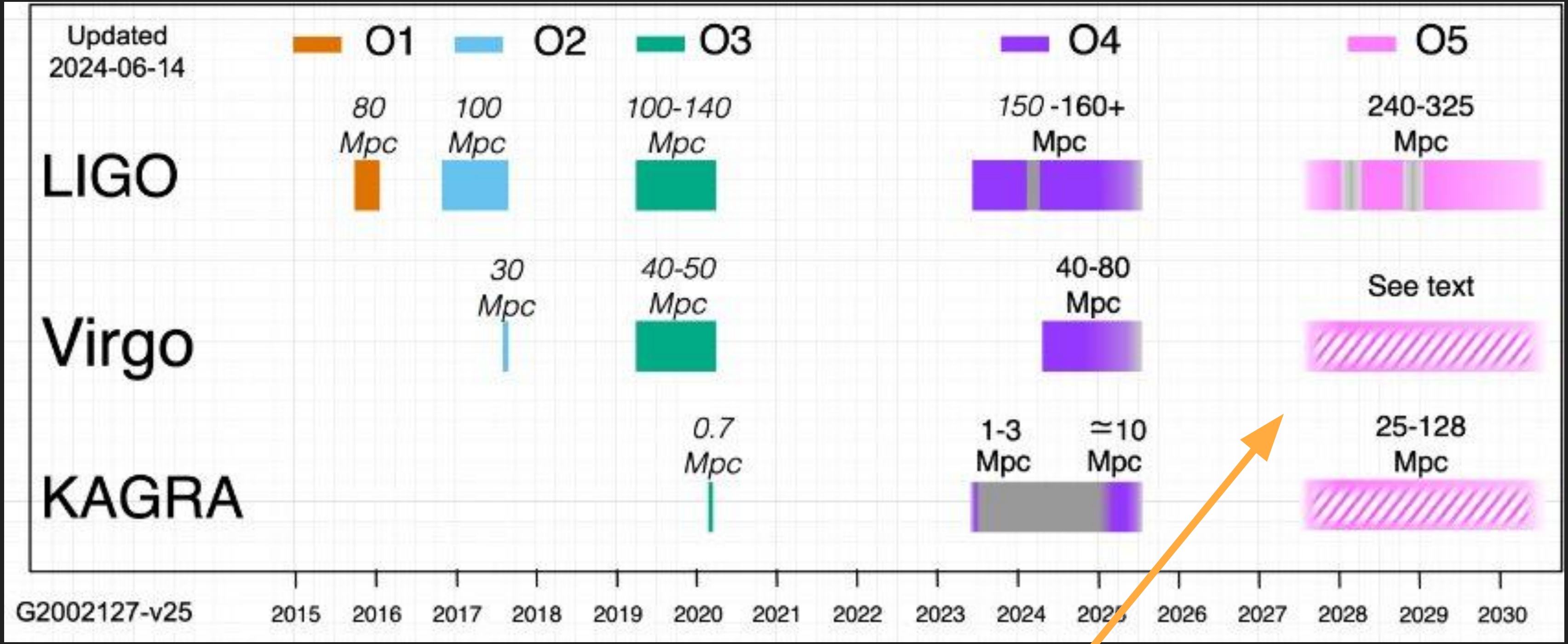
Parameter estimation for masses, spins, distance etc.



Population studies



The near future



Increased rate of detections - by a factor between 4 and 7. So increased computational requirements

Computing

Computing Demand

- LIGO, Virgo, KAGRA pool all resources
  - All available to all members
  - Planning, prioritisation, accounting, authorisation etc. is managed by a computing working group
- Demand has grown organically from principally HTC CPU offline workflows, to include low-latency, HTC GPU and some HPC CPU
  - This usually happens in bursts of activity

Computing Supply

- Previously:
  - Large, isolated HTC resources
  - Providers need a lot of effort to make resources uniform
  - Dedicated hardware for some needs
- Now:
  - Large isolated HTC pools
  - Massive distributed HTC pool
  - Multiple prioritisation layers

Computing Model 1: Low latency

As fast as possible production of approximate results

Low latency processing:

1. Distribution of instrumental data
2. Calibration and basic data quality checks
3. Distribution of calibrated data
4. Detection and significance estimates
5. Localisation, initial parameter estimation
6. Publication via alerts

How is this done?

1. Small set of dedicated resources at system level
2. As 1.
3. KAFKA to distribute custom data packets to many receivers
4. Same HTC pool as offline, but with super-high priority
5. As 4
6. Dedicated resources - mix of on-site, research cloud and commercial cloud

Computing Model 2: Offline Data and Software

Data Distribution

- Approx 1GB/hour == ~10TB/year
- File requests over HTTP
- Nearby caches of the data (see next slide), these fetch the data from the server, and cache it for next request
  - Regularly-accessed data is more easily available
- Want to move to a more intuitive model where this is managed better by HTCondor

Software distribution

- Can send software with the job and run in a container
- Centrally-managed software distributions distributed using CVMFS

BRIGHT SLIDE WARNING

LIGO
VIRGO
KAGRA

University of Glasgow

VIA VERITAS VITA

UNIVERSITY OF PORTSMOUTH

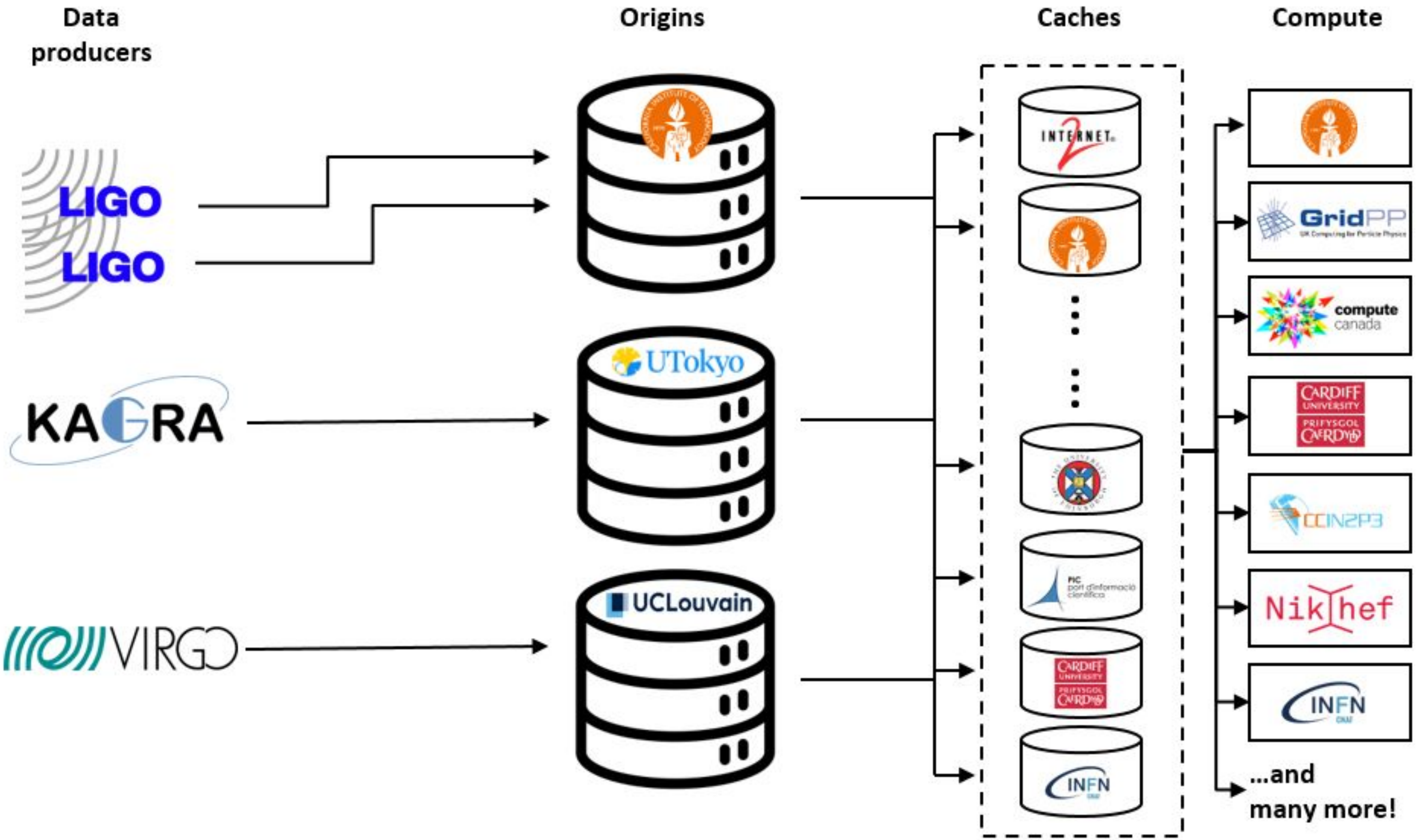


Diagram credit: Bagnasco (INFN)

Computing Model 3: Offline HTC

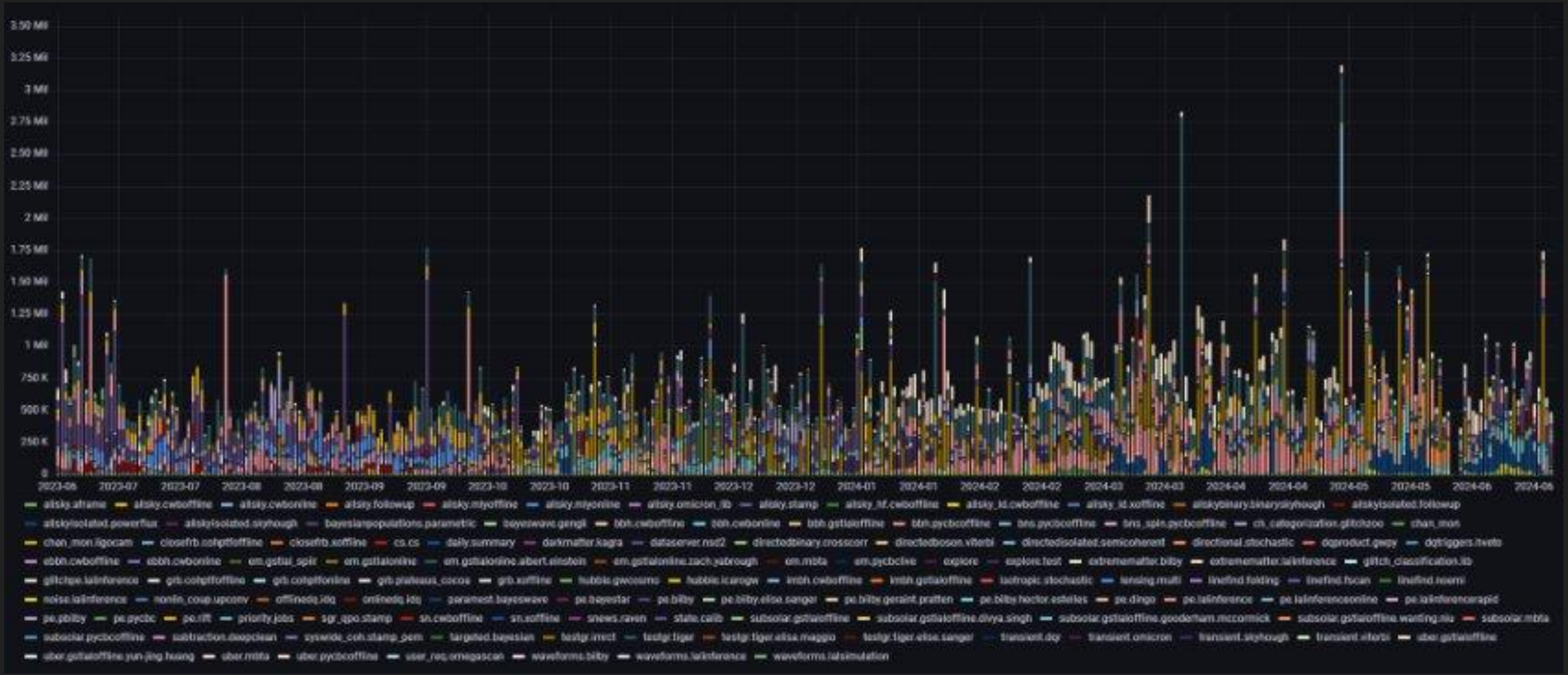
Standard model

Data access → super-parallelisable compute-intensive analysis → post-processing

We use the IGWN-grid infrastructure

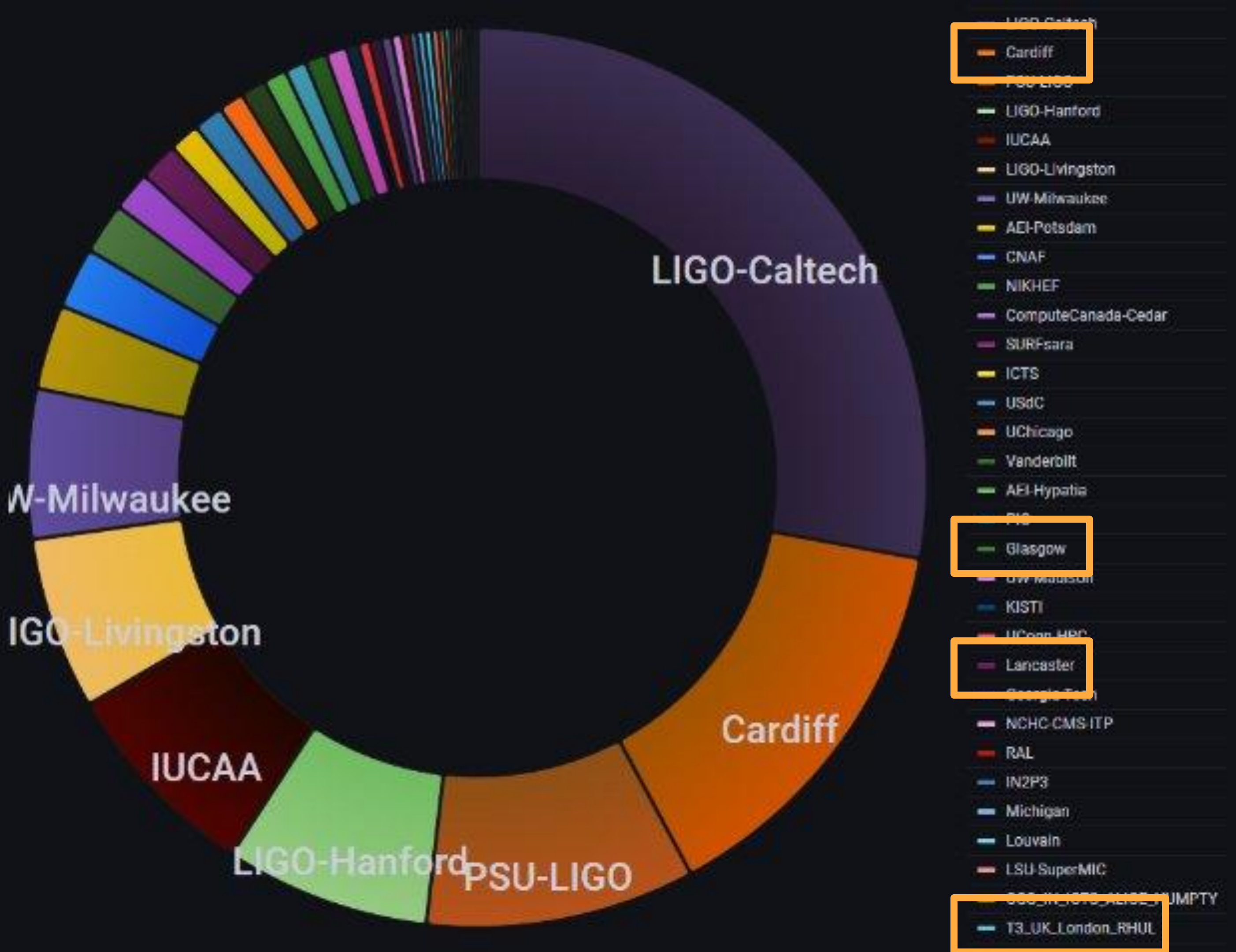
- Powered by HTCondor
- Multiple pools, all talking to a central factory to route jobs to their execute point
    - No need for submit nodes everywhere
    - Storage requirements are small
    - No need for local data copies
- A few homogeneous access points are needed
    - All users access all
    - Large persistent storage
    - Fast data access

How does demand vary over time?



Many different pipelines running!

Where do we use compute resources?



Computing Model 4: Offline HPC

- We don’t really do this
- Usually people use local HPC resources and run there
- This needs a solution for O5 with the greater number of events
- HTCondor for single-node jobs are mostly used
  - There is support more parallel jobs, but not many people know how to use this

Integration with GridPP / IRIS

- STFC supports UK LIGO contributions - including computing
- Dedicated compute UK LIGO commute resource in Cardiff (~5000 cores)
- Many IRIS providers (mainly GridPP) allow LVK jobs to run
- Desire to connect Cardiff cluster to IRIS/GridPP as a provider
    - Integrate accounting of UK LVK allocation/usage
    - Utilise any unused LVK resources for other IRIS workflows

Summary

- LVK Science:
  - Network of 4 detectors in operation
  - Observing run 4 ongoing until 2025
    - New results already, many more coming]
  - Observing run 5 scheduled for 2027
    - More sensitive = more detections = more compute requirements
- LVK computing
  - Dominated by CPU high throughput workflows
  - Distributed grid model
    - Remote data access
    - Software containers
  - Allows for (somewhat) fast integration / utilisation of new providers, including GridPP
- Strong desire for integration with IRIS and GridPP