Vulkanised 2025

The 7th Vulkan Developer Conference
Cambridge, UK | February 11-13, 2025

Compute shaders in astronomy: Vulkan for Gravitational Waves

Patrick Clearwater
Swinburne University of Technology

Who am I?

- Astrophysicist by training
  - (“astronomer” to impress people)
- Academic background: a PhD focused on gravitational wave data analysis
- But I also really like computers

What do I do now?

- **Astronomy Data and Computing Services (ADACS)**
- Funded by Australian government to provide software support to the Australian astronomy community
- Based across three Australian universities

ADACS
ASTRONOMY DATA AND COMPUTING SERVICES

Astronomy Australia Ltd.

NCRIS
National Research Infrastructure for Australia
An Australian Government Initiative

SWINBURNE UNIVERSITY OF TECHNOLOGY

Curtin University
THE CURTIN INSTITUTE FOR COMPUTATION

MACQUARIE University

What do I do now?

- **Astronomy Data and Computing Services**
- We do many things:
  - helping with software best practices, automated testing, …
  - training and documentation
  - web portals for astronomy codes
  - user experience research
  - **optimisation of existing code (incl with GPUs)**

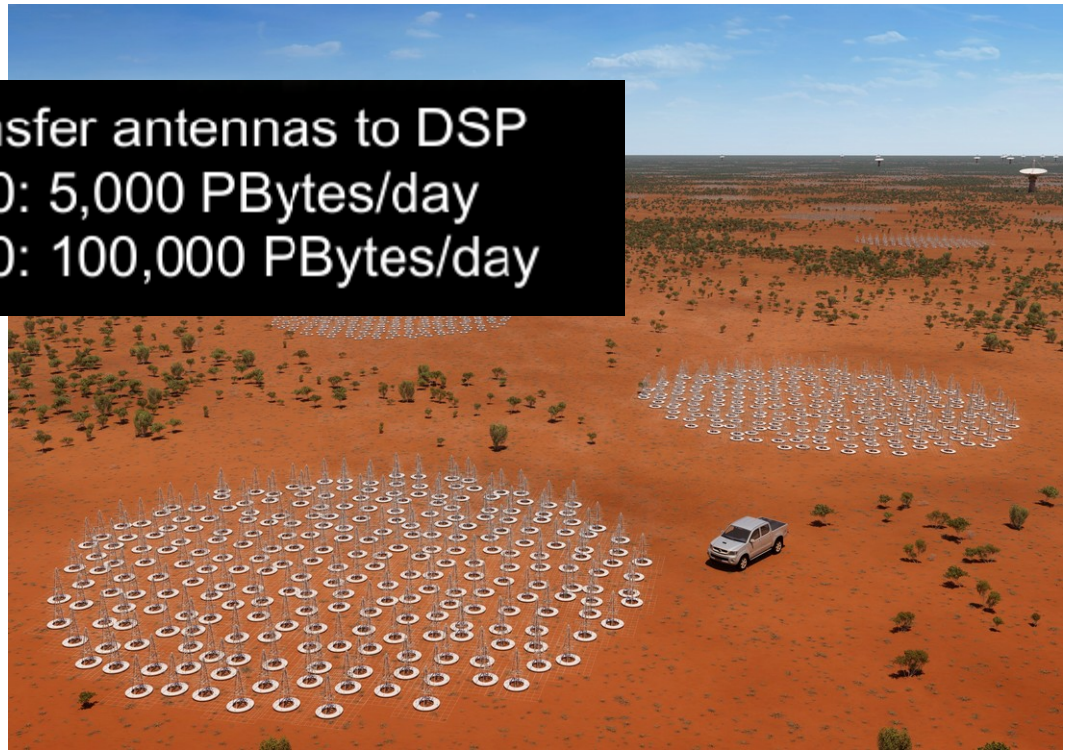
merit-based allocation process for access to ADACS resources

What do I do now?

- **Astronomy Data and Computing Services**
- Support university research translation efforts
- We also operate two research supercomputing clusters (free for Australian astronomy community)
- Also want to expand the “scientific software engineering service provider” ADACS model beyond astronomy

Computing in astronomy

- Modern astronomy is a data- and compute-intensive discipline
- Illustrative next-gen example: Square Kilometre Array



To Process in HPC
2020: 50 PBytes/day
2030: 10,000 PBytes/day

ref: Alexander/SKA, “The SKA Science Data Processor and Regional Centres” (2018)

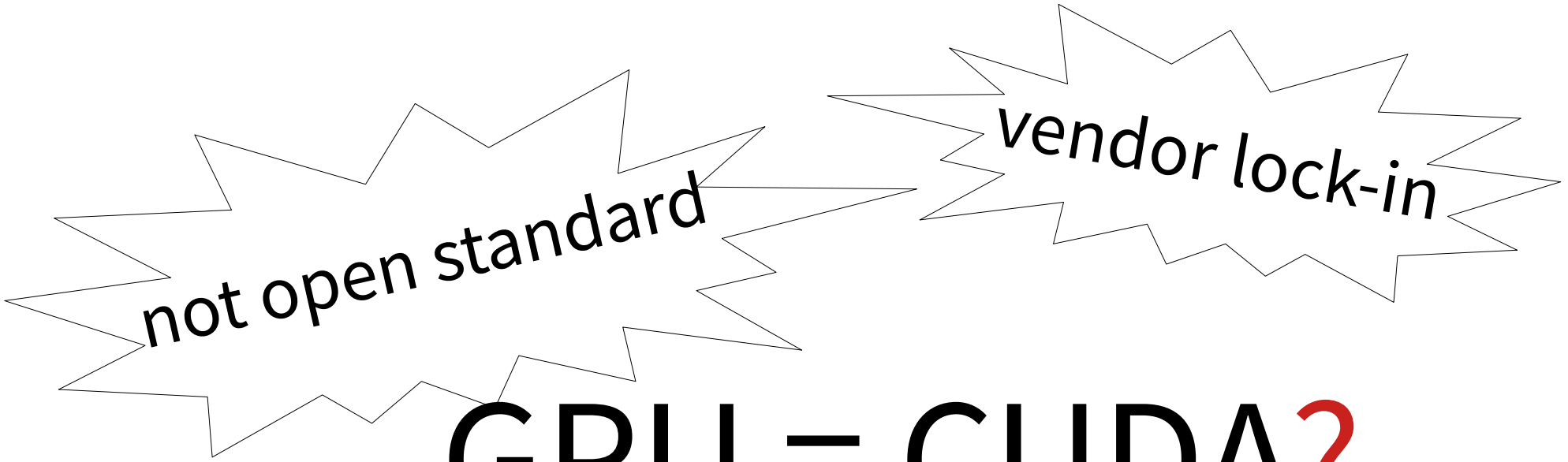
Artists’s impression: CSIRO

General-purpose GPUs in astronomy

- Obviously GPUs are of some interest
- Reasonably widely used for a long time

See e.g.
Barsdell et al “Analysing astronomy algorithms for graphics processing units and beyond”, *Monthly Notices of the Royal Astronomical Society* **408** 3 1936–1944 (2010)
Fluke et al “Astrophysical Supercomputing with GPUs: Critical Decisions for Early Adopters”, *Publications of the Astronomical Society of Australia* **28** 1 15–27 (2011)

GPU = CUDA



GPU = CUDA?

decision or default?

Alternatives?

- What alternatives exist?
- Can we use **Vulkan compute shaders** to do the same thing?

Vulkan case study

- My goal: port a simple, but real, astrophysical search code to Vulkan

ASTRONOMY PRIMER*
Gravitational waves

* These slides will not be on the exam

Gravitational waves

- Predicted by **general relativity**
- GR treats spacetime as geometry, which is deformed by matter

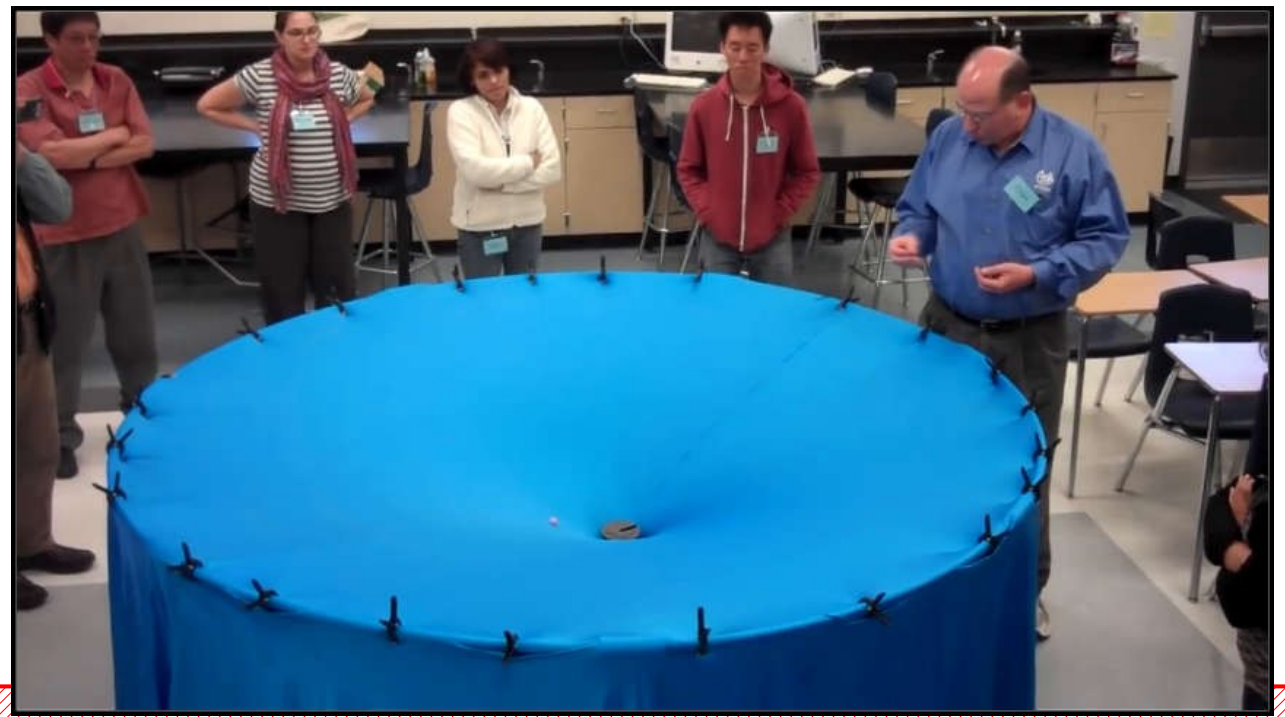
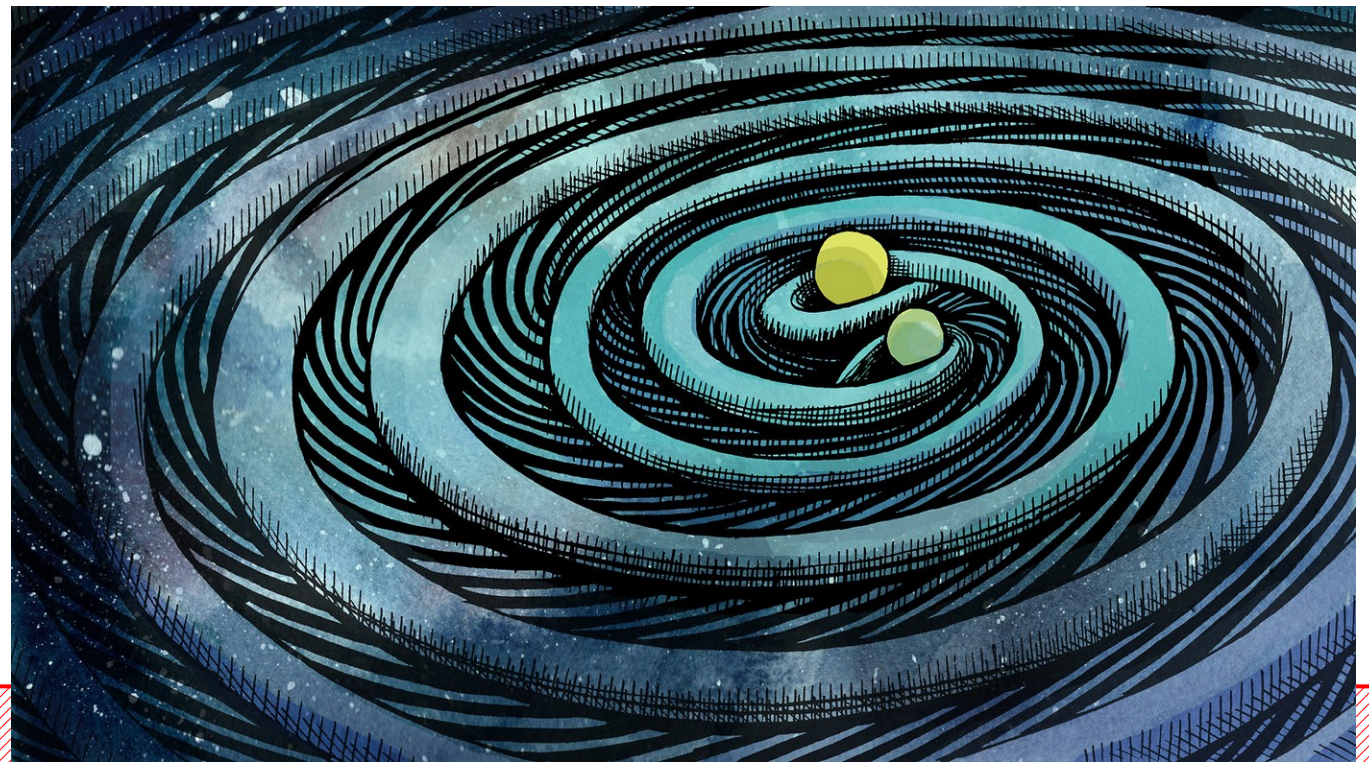


image: Dan Burns / PTSOS / Los Gatos High School

Gravitational waves

- Gravitational waves occur when an accelerating mass causes a wave to travel through space-time; like a pebble dropped in a pond



Artists's impression:
Corinne Mucha / Sandbox Studio

How do we search for them?

- Various ways — space based interferometer; pulsar timing
- Of interest here is the **International Gravitational Wave Observatory Network** (IGWN)
- Terrestrial interferometer network

How do we search for them?

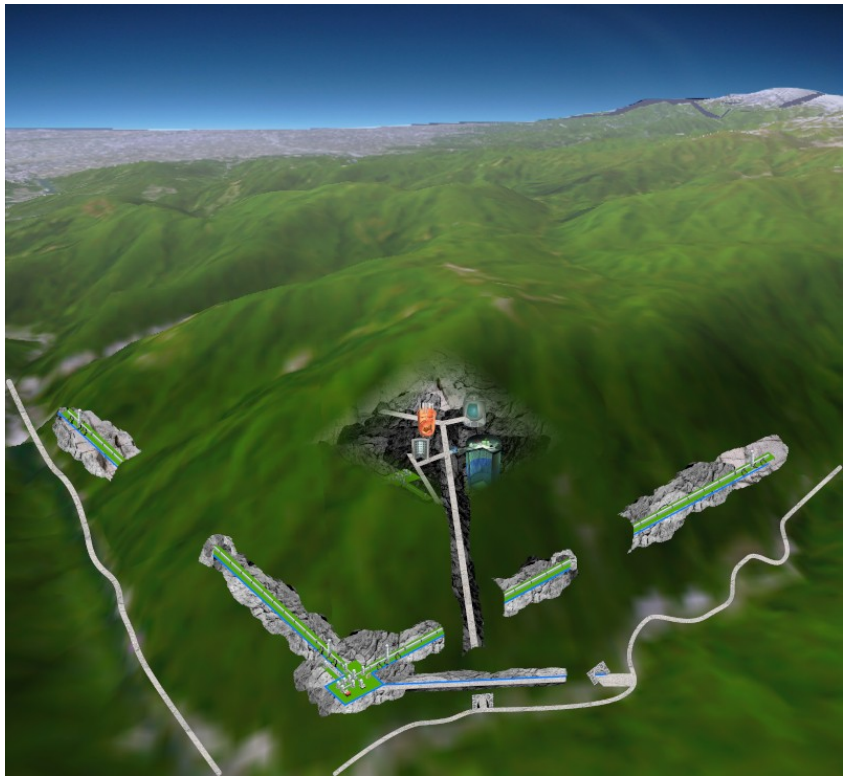
LIGO Livingston, Louisiana, U.S.

LIGO Hanford, Washington, U.S.

Images: LIGO

How do we search for them?

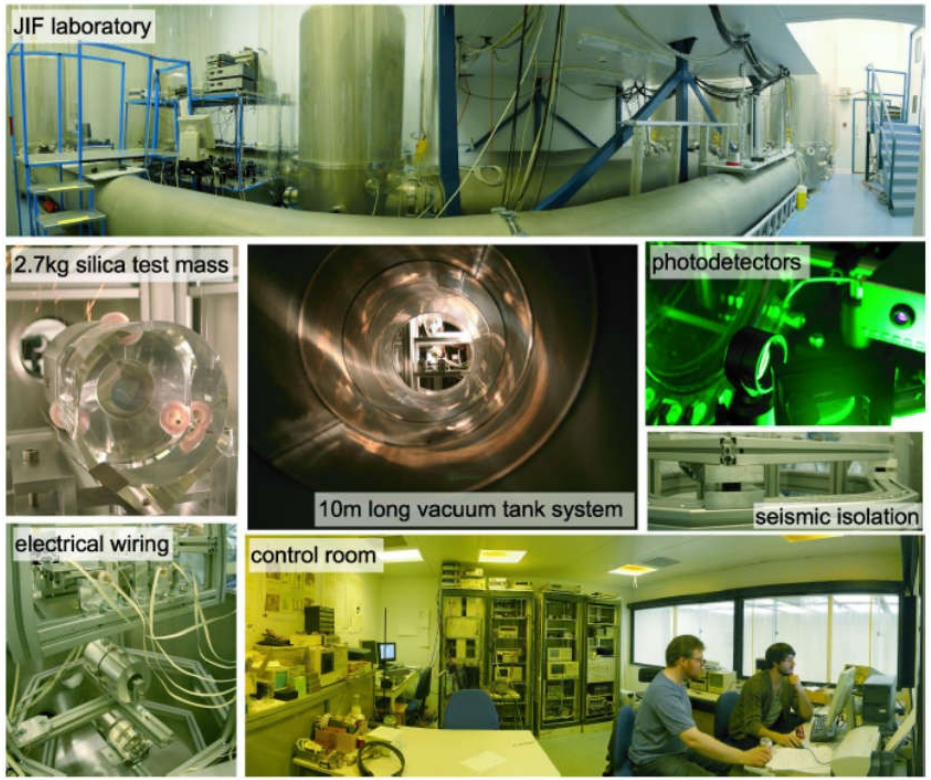
VIRGO, Cascina, Italy



KAGRA, Kamioka, Japan

Images: Virgo, Kagra/ICRR

Even in the UK…

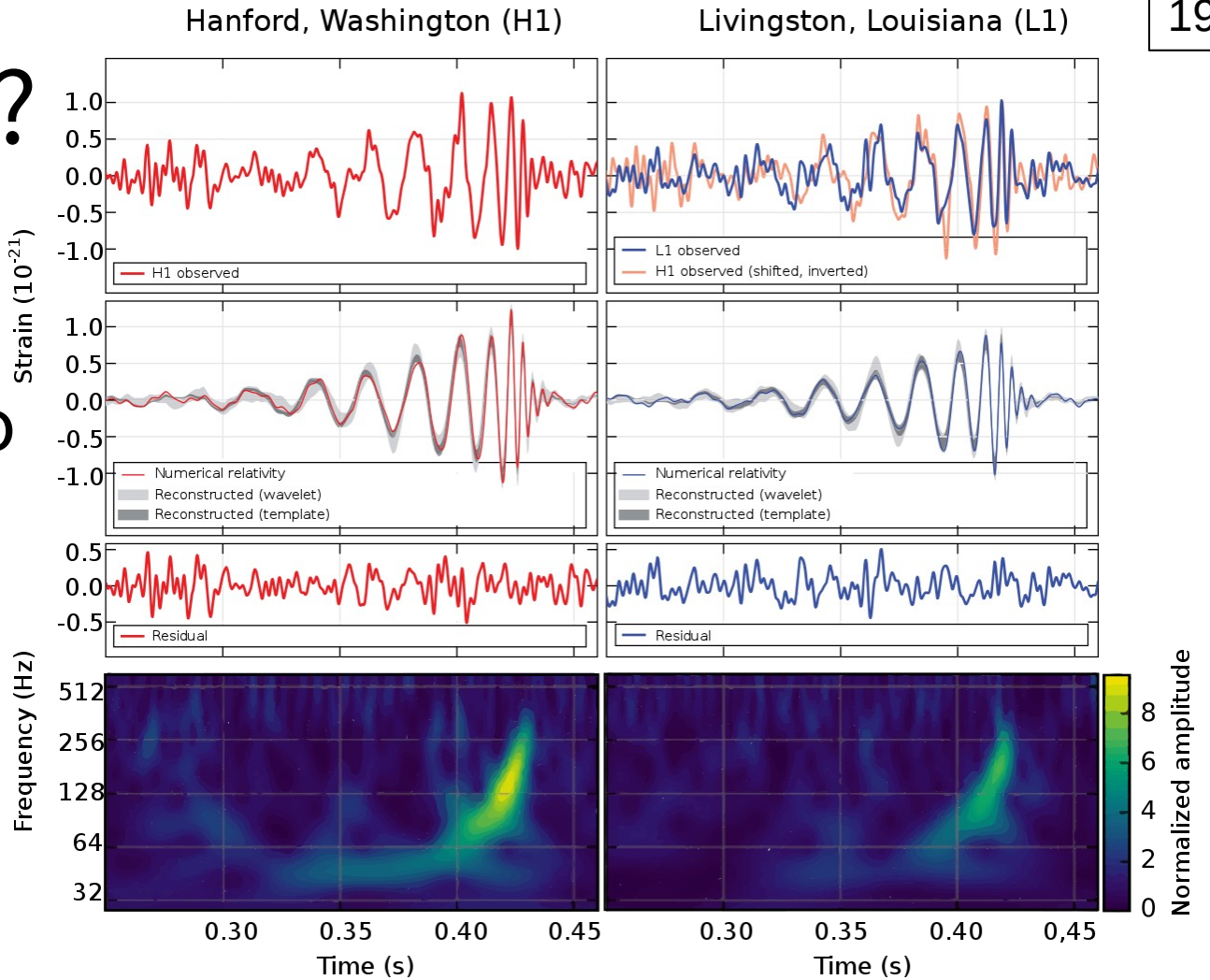


- Glasgow 10-metre interferometer, University of Glasgow
- (used for prototyping new technology; too small for astrophysical detections)

Image: S Hild, LIGO G-1000952

Are these even real?

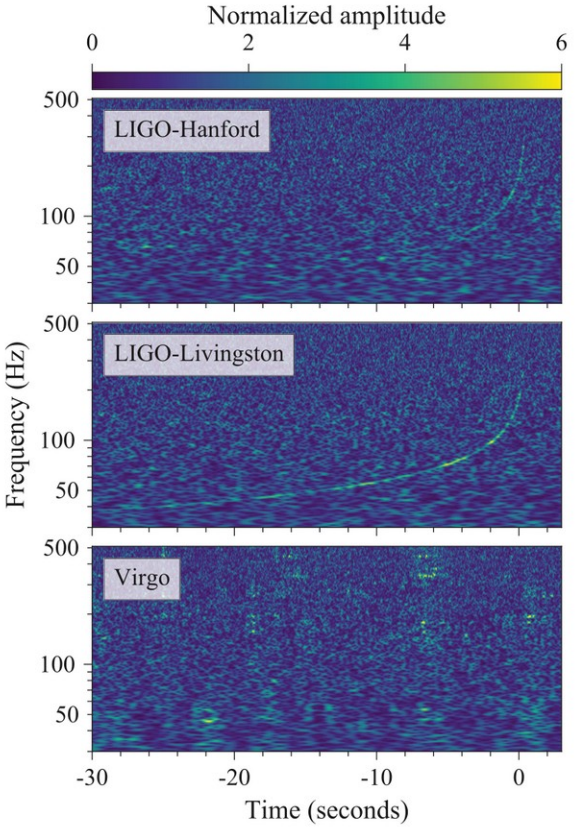
- Yes
- Detection in 2015 of two black holes merging (“GW150914”)



Abbott et al., *Physical Review Letters* **116** 061102 (2016)

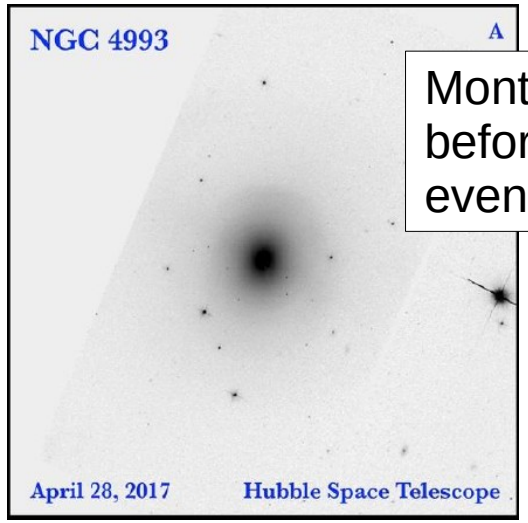
Detections

- Have also detected neutron star merger ("GW170817")
- In total approx ~150 "compact binary coalescence" detections

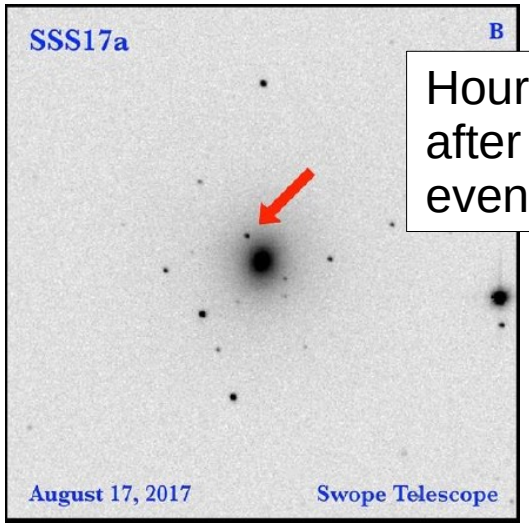


Abbott et al *PRL* **119** 161101 (2017)

HST; Apr 27 2017



Months before event

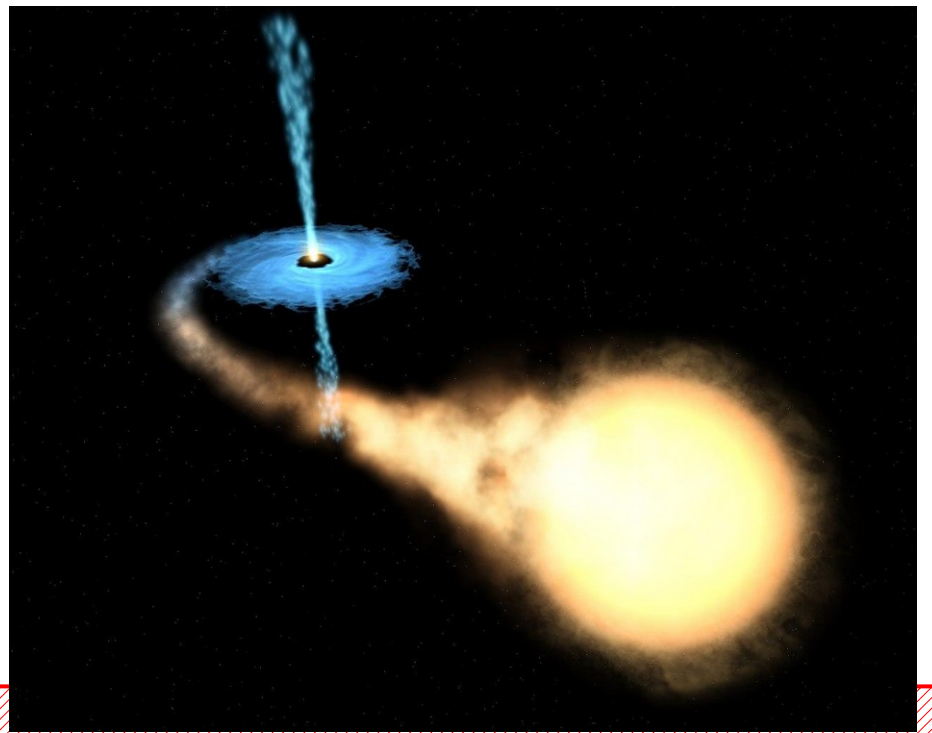


Hours after event

Coulter et al, **Science** 10.1126 /science.aap9811 [2017]

Continuous gravitational waves

- Forget all that, we'll search for something that hasn't been found yet… **continuous gravitational waves** from **neutron stars** in **low-mass x-ray binaries** (LMXBs)
- Present in data all the time (we hope) → so have to analyse a lot of data



Artist's impression: ESA

The code

- The code does (for our purposes) three things:
  1) accounts for the Earth's motion
  2) accounts for the binary star's motion
  3) tracks stochastic variation in intrinsic frequency

Algorithm described in: Suvorova, Clearwater, et al *Physical Review D* **96** 102006 (2017)

Algorithm and existing CUDA code used for example in:
- LIGO Scientific Collaboration, Abbott, et al PhysRevD **100** 122002 (2019) — Scorpius X-1 (brightest LMXB)
- Middleton, Clearwater et al, PhysRevD **102** 023006 (2022) — five other LMXBs
- LIGO Scientific Collaboration, Abbott et al, PhysRevD **105** 022002 (2022) — twenty pulsars
- LIGO Scientific Collaboration, Abbott et al, PhysRevD **106** 062002 (2022) — Scorpius X-1 (more data)

The code

has a CUDA version[1] but annoying to port for Vulkan for ecosystem reasons

- The code does (for our purposes) three things:
  1) accounts for the Earth's motion
  2) accounts for the binary star's motion
  3) tracks stochastic variation in intrinsic frequency

both are sensible candidates for a trial Vulkan port

[1] Dunn, Clearwater, Melatos, Wette *Classical and Quantum Gravity* **39** 045003 (2022)

Vulkan version

- We'll use Vulkan compute shaders of course
- But for style points:
  - write it in Rust
  - and write the shader in `slang`

Vulkan version

- And so I did that…
  - and it produced the right answers
  - and it was somewhat slower
  - and there was a lot more typing than CUDA

Lessons I

- A lot of boilerplate
  - … of course this is partly my fault

⇒ sensible to develop a framework or library for this

```rust
fn allocate_mem_gpu(
    device: &ash::Device,
    allocator: &mut gpu_allocator::vulkan::Allocator,
    buffer_size: usize,
    direction: gpu_allocator::MemoryLocation,
) -> Result<GpuAllocation> {
    let buffer = unsafe {
        device.create_buffer(
            &vk::BufferCreateInfo::default()
                .size(buffer_size as vk::DeviceSize)
                .usage(
                    vk::BufferUsageFlags::STORAGE_BUFFER
                        | vk::BufferUsageFlags::SHADER_DEVICE_ADDRESS,
                ),
            None,
        )?
    };

    let buffer_requirements = unsafe { device.get_buffer_memory_requirements(buffer) };

    let allocation = allocator.allocate(&gpu_allocator::vulkan::AllocationCreateDesc {
        name: "Input Buffer",
        requirements: buffer_requirements,
        location: direction,
        linear: true,
        allocation_scheme: gpu_allocator::vulkan::AllocationScheme::GpuAllocatorManaged,
    })?;

    unsafe { device.bind_buffer_memory(buffer, allocation.memory(), allocation.offset())? };

    return Ok(GpuAllocation {
        allocation,
        buffer,
        requirements: buffer_requirements,
    });
}
```

```c
int main() {
    double *the_thing;
    cudaMalloc(&the_thing, THE_SPACE);
}
```

Lessons II

- Accuracy and performance fine
  - Really this is to be expected — the code's not that complex and it's the same hardware

$\Rightarrow$ some attention needed with libraries

Lessons III

- Need to develop a real use-case for this
  - It was fun for me to do but real people need to get real things done

⇒ develop it, be ready

⇒ watch supercomputer refresh cycles

Next steps

- I did this for fun on my own time (thanks to Swinburne for letting me do it on their time, actually)
- Develop this into a more serious piece of code
- See if I can convince people to run it on a lot of different systems
- Go to conferences; tell people how cool astronomy is ✅

Summary

- CUDA is commonly used in astronomy to drive GPUs
- It’d be nice to use Vulkan compute shaders
- This works!
- Would love to hear from you

Patrick Clearwater
pclearwater@swin.edu.au