science & innovation
Department:
Science and Innovation
REPUBLIC OF SOUTH AFRICA

NRF National Research Foundation | SARAO South African Radio Astronomy Observatory

A PYTHON-BASED RADIO ASTRONOMY CORRELATOR

PyConZA 2022

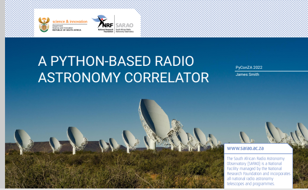
James Smith

www.sarao.ac.za

The South African Radio Astronomy Observatory (SARAO) is a National Facility managed by the National Research Foundation and incorporates all national radio astronomy telescopes and programmes.

A Python-based Radio Astronomy Correlator

2022-10-13



Outline

- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Outline
- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

First part of talk is going to be a radio astronomy history / information session, to give context of what a correlator is
Then I'll talk about what makes this one interesting,
Why Python has been helpful, and
Highlight an example of cool Python trick making it all possible.

Outline

- Radio Astronomy
  - Radio Astronomy in South Africa
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Outline
- Radio Astronomy
  - Radio Astronomy in South Africa
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

So what is Radio Astronomy?

The EM Spectrum

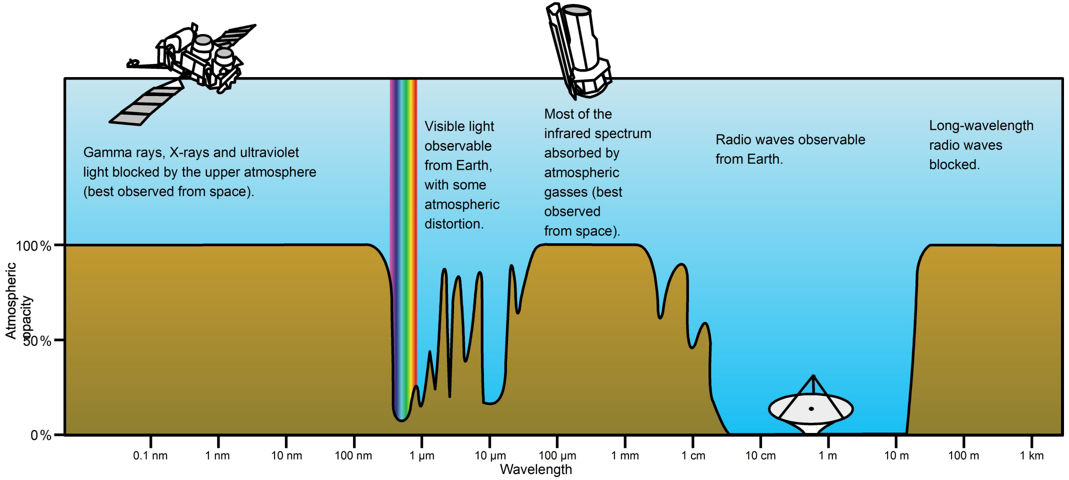
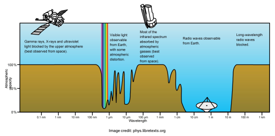


Image credit: phys.libretexts.org



More on the spectrum that just visible light.
Radio waves are much longer wavelength / lower frequency.

Outline

- Radio Astronomy

  Radio Astronomy in South Africa

- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Outline
- Radio Astronomy
Radio Astronomy in South Africa
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

South Africa has a history of doing radio astronomy, though it's from humble beginnings.

HartRAO

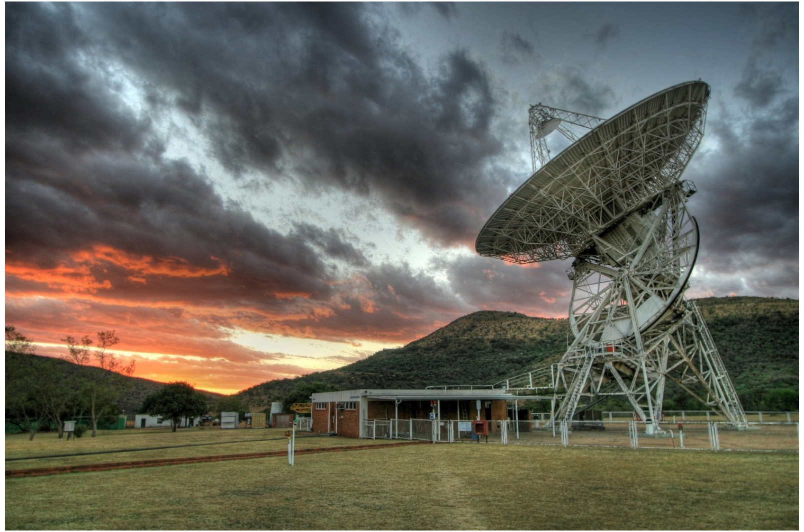
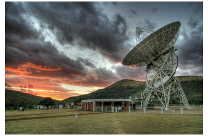


Image credit: Thomas Abbott / HartRAO

HartRAO



- NASA 26m ground station at Hartebeesthook
- 1974 handed over to South Africa
- Converted to radio astronomy facility
- Fairly important globally, for reasons that I'll get to
- Not much for 3 decades

XDM

Image credit: Mike Gaylard / HartRAO

XDM

Built another one in 2007
15m composite dish, named XDM
Reason we built it will be clear in a moment.

Bigger is Better

Image credit: NRAO/AUI/NSF

Bigger is better
Bigger is also exponentially more expensive
This is Green Bank in West Virginia
Largest fully-steerable telescope in the world

Outline

- Radio Astronomy
- Interferometers
  - SKA Precursors
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

But you don’t just need to use one dish at a time.
Lots of little dishes.

This falls under the broader scientific discipline called Interferometry.

Outline
Radio Astronomy
Interferometers
SKA Precursors
Correlators
Maths
Shiny new tech
katgpucbf
Python
Conclusion

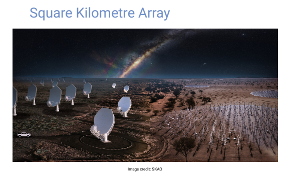
Basic Concept

Image credit: NRAO/AUI/NSF

LA (Very Large Array) in New Jersey, USA.
Signals combined and the individual antennas function together as one telescope.

Square Kilometre Array

Image credit: SKAO



Back in the 90s -> square kilometre collecting area.
Needed a good home-base.
South Africa bid for hosting.
XDM built to prove to ourselves that we could do it.
But it was just a single antenna, we needed interferometers.

Outline

- Radio Astronomy
- Interferometers
  - SKA Precursors
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Outline
Radio Astronomy
Interferometers
SKA Precursors
Correlators
Maths
Shiny new tech
katgpucbf
Python
Conclusion

Both South Africa and Australia built precursors to demonstrate the suitability of the site and our own technical capabilities.

KAT-7

Image credit: SARAO

The first is called KAT-7. Karoo Array Telescope, with 7 elements.

MeerKAT

Image credit: SARAO

We then went on to build MeerKAT. Die naam is eenvoudig. Meer KAT. The fact that you do get meercats in the Karoo area is completely coincidental.

MeerKAT Extension

Coming soon...

MeerKAT Extension
Coming soon...

And very soon, we'll be extending it. This is the project that I'm working on at the moment. And specifically...

Outline

- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Outline
- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

This is the name for the machines that combine signals from multiple telescopes into one.

Not much to look at

Image credit: NRAO/AUI/NSF

Not much to look at

At this point the pictures start to get somewhat less interesting.
First VLA correlator.
Best kind - "real-time."
Unified output, useful for science, as soon as observation ends.
Usually means dedicated hardware.
This usually means that dedicated hardware needs to be built.

Need a hard drive?

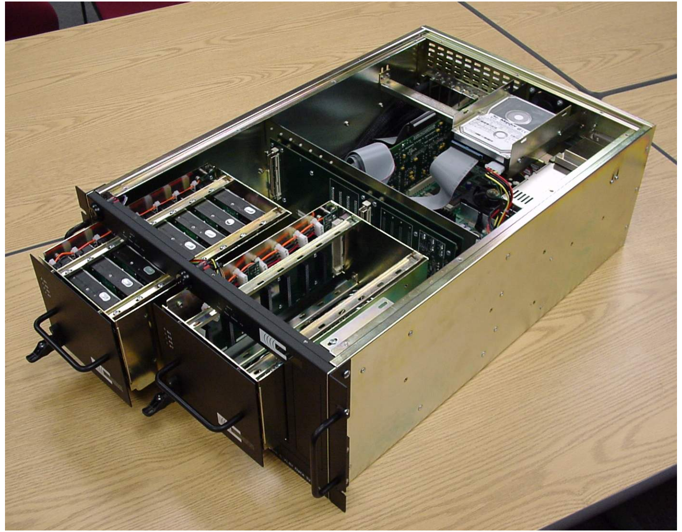
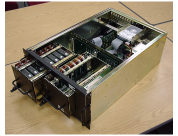


Image credit:

Need a hard drive?



Up to now, software-correlators were slow.
This isn't a correlator, this is a recorder used at VLBI stations.
VLBI is why HartRAO is important internationally.
Data recorded directly to disk, then transported to a central location
Correlation is done by software in batches, can take a long time.

FPGA boards

Image credit: Peralex

The South African precursors have real-time correlators
FPGA-based hardware
Can be re-flashed with new signal-processing logic as the designs are refined
Not as fast as re-deploying new software, as FPGA logic is challenging
but certainly not as challenging as re-building hardware.
Currently around 280 such SKARABs in the Karoo.

Outline

- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Outline
- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Phase

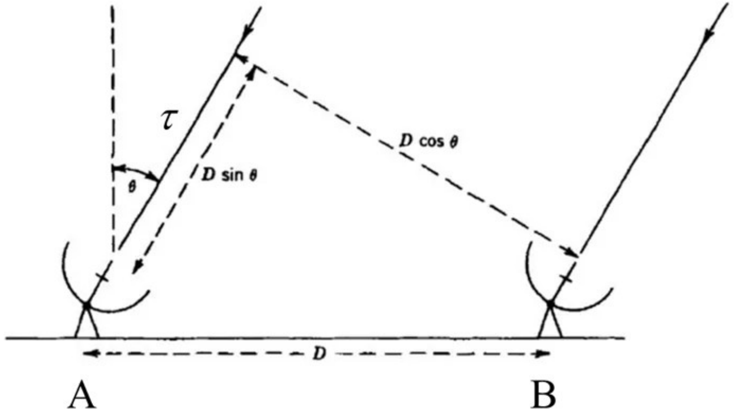
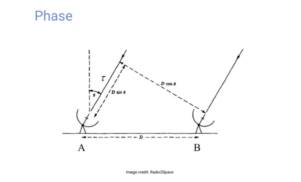


Image credit: Radio2Space



Consider pair of antennas, pointed in same direction.
One will get signal wavefront first, tau time delay.
Why this works is interesting by itself, but that tau (or phase) has most of interesting astronomical information.

Two critical concepts to understand.

Multiplication

$\times$

Multiplication.
Ok specifically complex conjugate multiplication. But it's the same.
Trig identity - product of sines, output proportional to difference.
A bit more complicated but that will get you in the right direction.

Multiplication

×

Addition

$+$

Addition

+

Addition

Like a long exposure camera, to increase SNR.

That's it. Now you know what you need in order to be able to code up a quick correlator of your own at home.

Fourier transform

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \mathrm{e}^{-\mathrm{j}2\pi\omega t} \, \mathrm{d}t \; \forall \, \omega \in \mathbb{R}$$

Fourier transform

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \mathrm{e}^{-\mathrm{j}2\pi\omega t} \, \mathrm{d}t \; \forall \, \omega \in \mathbb{R}$$

I may have oversimplified things ever so slightly.
What I've just mentioned is really only true in the case of narrow bandwidth signals.
Modern radio telescopes have wideband receivers.
Fourier transformation allows us to decompose a wideband signal into many narrowband ones.
Fortunately there are libraries for this one, and it's quite fast.

Problems

So why is specialised hardware so often required?

Problems

So why is specialised hardware so often required?

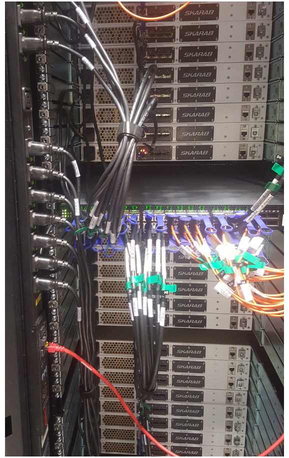
not computational intensity of the data, but interconnect.
Data needs to get onto and off the compute nodes fast enough to be useful.

Outline

Outline
- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

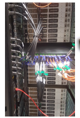
Learned since we started building telescopes - leverage commercial tech where possible.
In the past - not always possible. Recently? Yes!
For MeerKAT Extension, we have real-time processing capability, but software.
This is how.

Ethernet



Rather than designing backplanes, we can use switches.

Ethernet



Rather than designing backplanes, we can use switches.

Interconnect between processing nodes themselves.
KAT-7 used 10GbE, and MeerKAT used 40GbE.

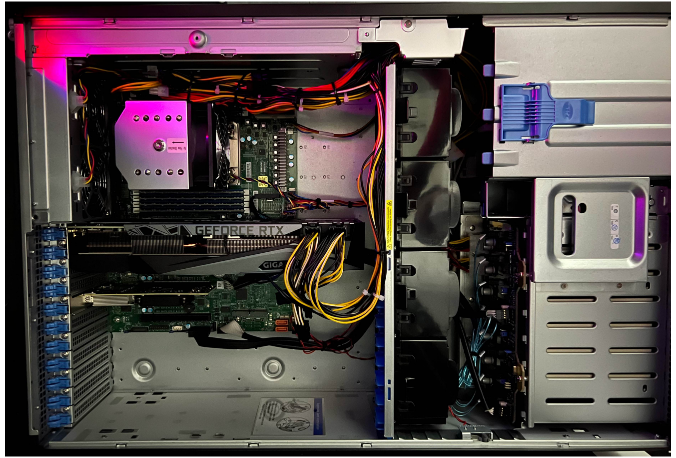
PCIe 4.0

Transferring data from network to accelerator.

PCIe 4.0
Transferring data from network to accelerator.

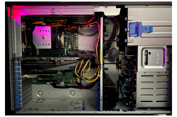
Interconnect within the processing nodes.
MeerKAT's SKARABs have their 40 GbE network interfaces wired right into the FPGAs.
In a computer, everything goes via system RAM and often the CPU.
4th generation PCIe and DDR4, this is finally quick enough to be useful.

GPUs



Everyone’s favourite computer upgrade.

GPUs



Everyone’s favourite computer upgrade.

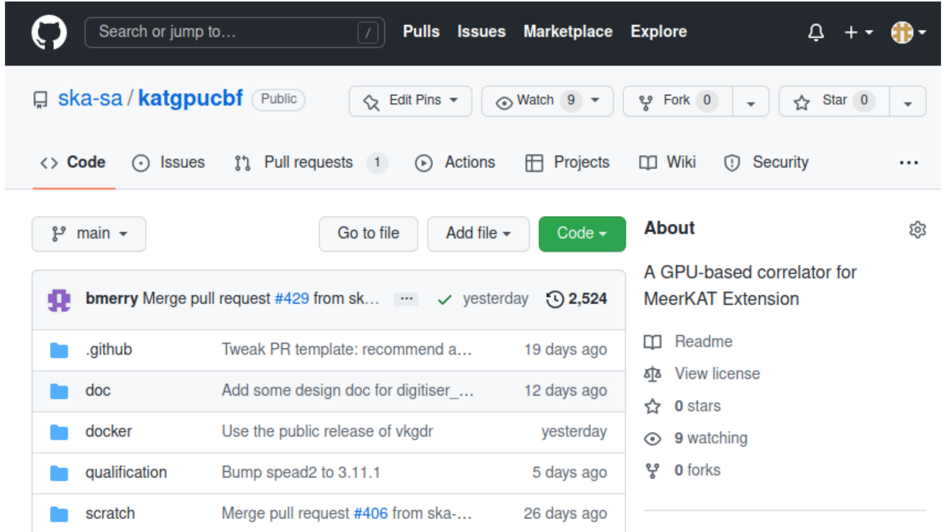
GPUs have actually been fast enough for quite a while.
Concept was prototyped on a Geforce GTX 1000-series card.
RTX 3000-series cards are PCIe 4.0.

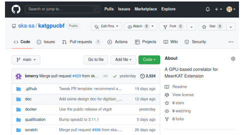
Outline

- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Outline
- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

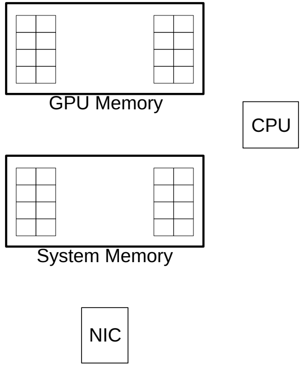
Module

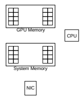




So we made this!
all open-source, so please try it at home.

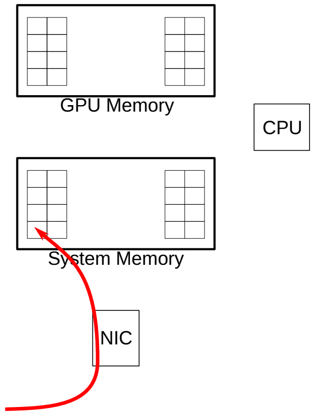
How it works



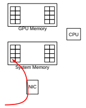


How it works

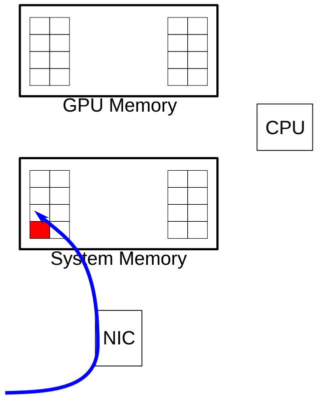
How it works

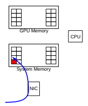


How it works



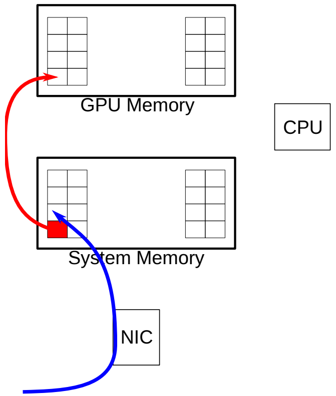
How it works





How it works

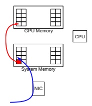
How it works



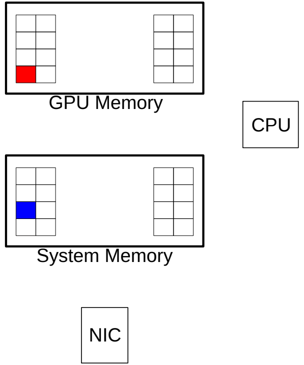
2022-10-13

A Python-based Radio Astronomy Correlator
└katgpucbf
└How it works

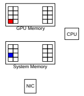
How it works



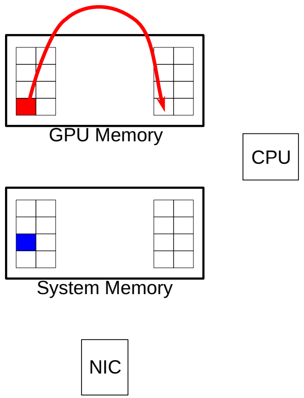
How it works



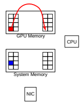
How it works



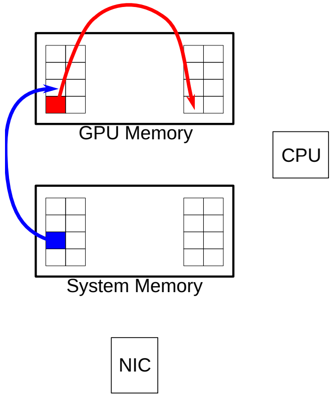
How it works

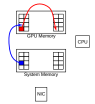


How it works

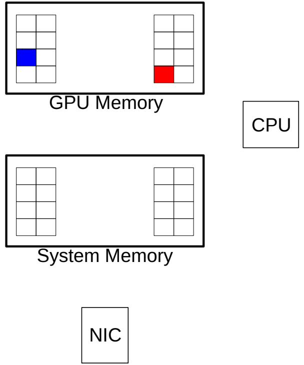


How it works

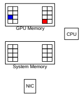




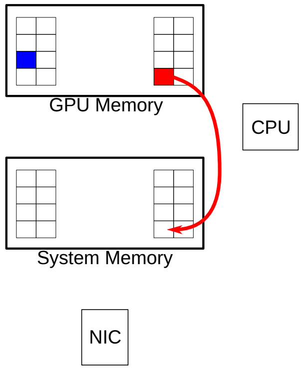
How it works



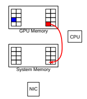
How it works



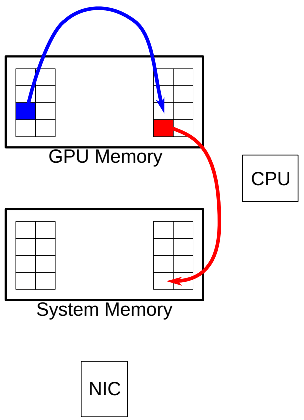
How it works

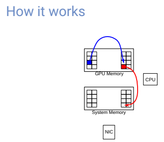


How it works

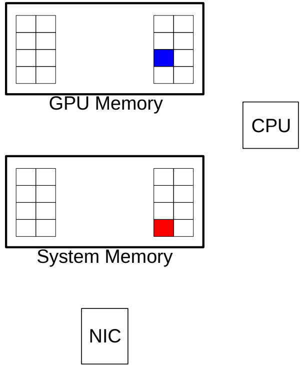


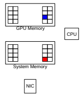
How it works



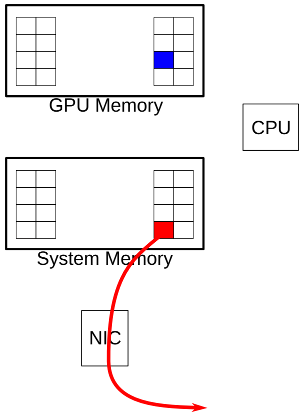


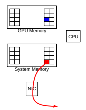
How it works



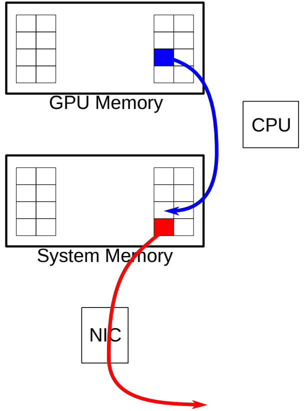


How it works





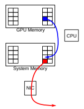
How it works



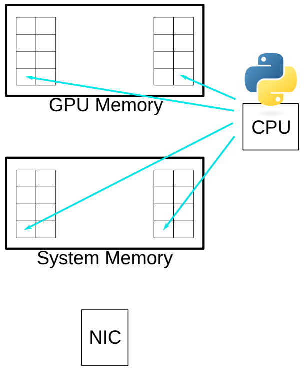
2022-10-13

A Python-based Radio Astronomy Correlator
└─katgpucbf
└─How it works

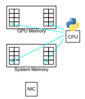
How it works



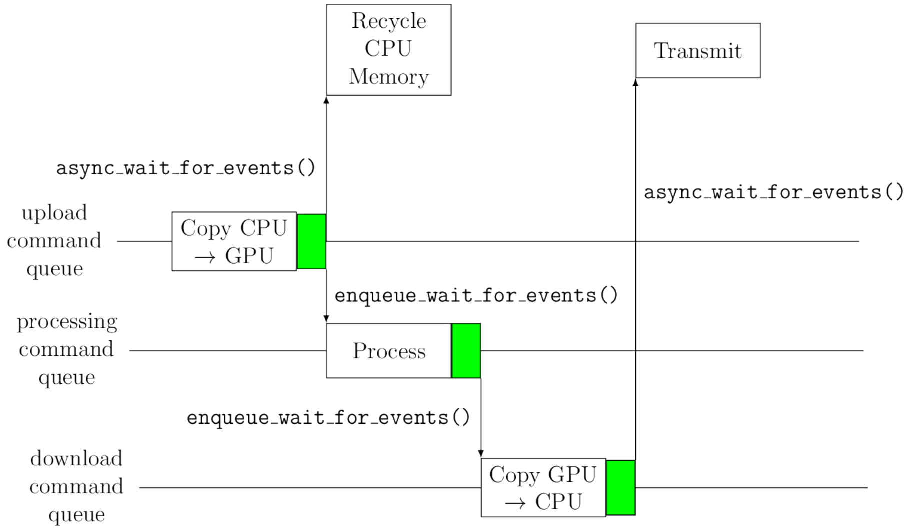
How it works



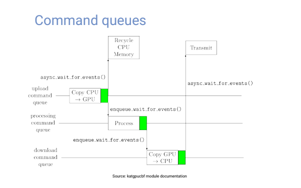
How it works



Command queues



Source: katgpucbf module documentation



Outline

- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- **Python**
- Conclusion

Outline
- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Handy Python modules

- asyncio

Handy Python modules

- asyncio

1. coordinating the data transfers and processing actions

Handy Python modules

- asyncio
- Buffer protocol

Handy Python modules
- asyncio
- Buffer protocol

1. coordinating the data transfers and processing actions
2. allocating and manipulating all the buffers
   Numpy arrays are a good example of this in action.

Handy Python modules

- asyncio
- Buffer protocol
- PyCUDA

Handy Python modules
- asyncio
- Buffer protocol
- PyCUDA

1. coordinating the data transfers and processing actions
2. allocating and manipulating all the buffers
   Numpy arrays are a good example of this in action.
3. getting the GPU out of bed
   plays nicely with numpy and aforementioned buffer protocol.

Handy Python modules

- asyncio
- Buffer protocol
- PyCUDA
- numba

Handy Python modules

- asyncio
- Buffer protocol
- PyCUDA
- numba

1. coordinating the data transfers and processing actions
2. allocating and manipulating all the buffers
   Numpy arrays are a good example of this in action.
3. getting the GPU out of bed
   plays nicely with numpy and aforementioned buffer protocol.
4. JIT compiles.
   Advantageous for speed, but also for interfaceing with modules written in C / C++.
   Will discuss further in a moment.

Handy Python modules

- asyncio
- Buffer protocol
- PyCUDA
- numba
- pytest

Handy Python modules

- asyncio
- Buffer protocol
- PyCUDA
- numba
- pytest

1. coordinating the data transfers and processing actions
2. allocating and manipulating all the buffers
   Numpy arrays are a good example of this in action.
3. getting the GPU out of bed
   plays nicely with numpy and aforementioned buffer protocol.
4. JIT compiles.
   Advantageous for speed, but also for interfaceing with modules written in C / C++.
   Will discuss further in a moment.
5. And then of course, testing. I won’t say more on this now, I’ll talk on this in more detail tomorrow.

Handy Python modules

- asyncio
- Buffer protocol
- PyCUDA
- numba
- pytest

Handy Python modules
- asyncio
- Buffer protocol
- PyCUDA
- numba
- pytest

Rolling our own

Functionality shared with other SARAO-internal software projects.
Much is open-source, so others could use if they want.

Rolling our own

- katsdpsigproc

Rolling our own

- katsdpsigproc

Functionality shared with other SARAO-internal software projects.
Much is open-source, so others could use if they want.

1. Signal Processing.
   wrapper around PyCUDA or PyOpenCL
   handles a lot of the boiler-plate stuff, general quality-of-life improvements for working with GPUs.

Rolling our own

- katsdpsigproc
- spead2

Rolling our own

- katsdpsigproc
- spead2

Functionality shared with other SARAO-internal software projects.
Much is open-source, so others could use if they want.

1. Signal Processing.
   wrapper around PyCUDA or PyOpenCL
   handles a lot of the boiler-plate stuff, general quality-of-life improvements for working with GPUs.
2. implemenation of an in-house data streaming protocol,
   used over the high-speed network for radio data.
   facilitates low-level access to networking hardware
   written in C++, with python bindings.

Rolling our own

- katsdpsigproc
- spead2
- vkgdr

Rolling our own

- katsdpsigproc
- spead2
- vkgdr

Functionality shared with other SARAO-internal software projects.
Much is open-source, so others could use if they want.

1. Signal Processing.
   wrapper around PyCUDA or PyOpenCL
   handles a lot of the boiler-plate stuff, general quality-of-life improvements for working with GPUs.
2. implemenation of an in-house data streaming protocol,
   used over the high-speed network for radio data.
   facilitates low-level access to networking hardware
   written in C++, with python bindings.
3. Makes use of the Vulkan API to map GPU's memory into the host's memory address space, to speed up transfers.

Rolling our own

- katsdpsigproc
- spead2
- vkgdr
- aiokatcp

Rolling our own
- katsdpsigproc
- spead2
- vkgdr
- aiokatcp

Functionality shared with other SARAO-internal software projects.
Much is open-source, so others could use if they want.

1. Signal Processing.
   wrapper around PyCUDA or PyOpenCL
   handles a lot of the boiler-plate stuff, general quality-of-life improvements for working with GPUs.
2. implemenation of an in-house data streaming protocol,
   used over the high-speed network for radio data.
   facilitates low-level access to networking hardware
   written in C++, with python bindings.
3. Makes use of the Vulkan API to map GPU's memory into the host's memory address space, to speed up transfers.
4. Protocol by which subsystems talk to each other for control and monitoring
   workis well with asyncio.

Rolling our own

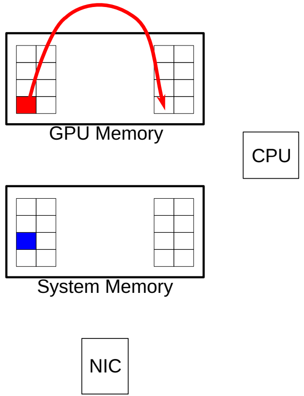
- katsdpsigproc
- spead2
- vkgdr
- aiokatcp

Rolling our own

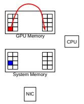
- katsdpsigproc
- spead2
- vkgdr
- aiokatcp

Functionality shared with other SARAO-internal software projects.
Much is open-source, so others could use if they want.

Large work chunks



Large work chunks



For a GPU to be even remotely efficient processing stuff, you need lots of parallelism.

Each block of order hundreds of megabytes, certainly much larger than even a Jumbo network packet (around 9kB).

Got to build up those chunks somehow.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.
- spead2 module (C++) doesn't know how the user wants to arrange the packet payload in memory, caller needs to provide logic.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.
- spead2 module (C++) doesn't know how the user wants to arrange the packet payload in memory, caller needs to provide logic.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.
- spead2 module (C++) doesn't know how the user wants to arrange the packet payload in memory, caller needs to provide logic.
- pybind11 is icky so we'd prefer to avoid it if possible.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.
- spead2 module (C++) doesn't know how the user wants to arrange the packet payload in memory, caller needs to provide logic.
- pybind11 is icky so we'd prefer to avoid it if possible.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.
- spead2 module (C++) doesn't know how the user wants to arrange the packet payload in memory, caller needs to provide logic.
- pybind11 is icky so we'd prefer to avoid it if possible.
- Passing each individual packet up into Python space for placement would be far too slow.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.
- spead2 module (C++) doesn't know how the user wants to arrange the packet payload in memory, caller needs to provide logic.
- pybind11 is icky so we'd prefer to avoid it if possible.
- Passing each individual packet up into Python space for placement would be far too slow.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.
- spead2 module (C++) doesn't know how the user wants to arrange the packet payload in memory, caller needs to provide logic.
- pybind11 is icky so we'd prefer to avoid it if possible.
- Passing each individual packet up into Python space for placement would be far too slow.
- So we define the logic in friendly, familiar Python, then JIT compile it so that the C++ module can use it.

Chunk Placement

- Network traffic is UDP, may arrive out-of-order, especially if from multiple sources.
- Individual packets contain metadata indicating where they come from and what they contain - anntenna, timestamp, frequency, etc.
- spead2 module (C++) doesn't know how the user wants to arrange the packet payload in memory, caller needs to provide logic.
- pybind11 is icky so we'd prefer to avoid it if possible.
- Passing each individual packet up into Python space for placement would be far too slow.
- So we define the logic in friendly, familiar Python, then JIT compile it so that the C++ module can use it.

Make a callback

```
@numba.cfunc(types.void(types.CPointer(chunk_place_data), types.uintp),
    nopython=True)
def chunk_place(data_ptr, data_size):
    data = numba.carray(data_ptr, 1)
    items = numba.carray(intp_to_voidptr(data[0].items), 2, dtype=np.
        int64)
    heap_cnt = items[0]
    payload_size = items[1]
    # If the payload size doesn't match, discard the heap (could be
        descriptors etc).
    if payload_size == HEAP_PAYLOAD_SIZE:
        data[0].chunk_id = heap_cnt // HEAPS_PER_CHUNK
        data[0].heap_index = heap_cnt % HEAPS_PER_CHUNK
        data[0].heap_offset = data[0].heap_index * HEAP_PAYLOAD_SIZE
```

Source: spead2 module examples folder

Make a callback

I'm only going to show a little bit of source code, this slide and the next one.
Don't worry too much about the logic, it's available in spead2's examples.
First few lines give you handles for various packet metadata.
Last few lines tell you where to place the packet payload based on that metadata.
Top line is a decorator which tells Numba to compile it.

Pass it on

```python
place_callback = scipy.LowLevelCallable(
    chunk_place.ctypes,
    signature='void (void *, size_t)'
)
chunk_config = spead2.recv.ChunkStreamConfig(
    items=[spead2.HEAP_CNT_ID, spead2.HEAP_LENGTH_ID],
    max_chunks=MAX_CHUNKS,
    place=place_callback)
```

Source: spead2 module examples folder

Pass it on

Then we use a convenience function provided by Scipy to get access to the function pointer, which we pass to the spead2 library which can use the funtion to arrange payload data.

Pros and cons

- Unfortunately C++ still exists.

Pros and cons

- Unfortunately C++ still exists.

1. Writing this logic doesn’t eliminate the need to be able to think in a C++ fashion, because you need to know about function signatures, function pointers, normal pointers, things like that.

Pros and cons

- Unfortunately C++ still exists.
- But at least we don't have to compile it while installing.

Pros and cons

- Unfortunately C++ still exists.
- But at least we don't have to compile it while installing.

1. Writing this logic doesn't eliminate the need to be able to think in a C++ fashion, because you need to know about function signatures, function pointers, normal pointers, things like that.

Pros and cons

- Unfortunately C++ still exists.
- But at least we don't have to compile it while installing.
- And we can benefit from the speed!

Pros and cons

- Unfortunately C++ still exists.
- But at least we don't have to compile it while installing.
- And we can benefit from the speed!

Outline

- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Outline
- Radio Astronomy
- Interferometers
- Correlators
- Maths
- Shiny new tech
- katgpucbf
- Python
- Conclusion

Take-aways

- Outsource hardware development where possible.

Take-aways

- Outsource hardware development where possible.

1. Economies of scale are a thing, not just in terms of cost. Companies dedicated to make e.g. GPUs will make a much more stable and mature product than a bunch of astronomers trying to get something to work well enough for them to do some science.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!

1. Economies of scale are a thing, not just in terms of cost. Companies dedicated to make e.g. GPUs will make a much more stable and mature product than a bunch of astronomers trying to get something to work well enough for them to do some science.
2. In no small part, anyway. Making use of stable mature products developed elsewhere applies to software tools too.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.

1. Economies of scale are a thing, not just in terms of cost. Companies dedicated to make e.g. GPUs will make a much more stable and mature product than a bunch of astronomers trying to get something to work well enough for them to do some science.
2. In no small part, anyway. Making use of stable mature products developed elsewhere applies to software tools too.
3. Despite these things, MKE CBF built in shorter time and with fewer engineers and less money than MK. Fingers crossed.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.
  - Pandemic supply-chain issues

Take-aways
- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.
  - Pandemic supply-chain issues

1. Economies of scale are a thing, not just in terms of cost. Companies dedicated to make e.g. GPUs will make a much more stable and mature product than a bunch of astronomers trying to get something to work well enough for them to do some science.
2. In no small part, anyway. Making use of stable mature products developed elsewhere applies to software tools too.
3. Despite these things, MKE CBF built in shorter time and with fewer engineers and less money than MK. Fingers crossed.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.
  - Pandemic supply-chain issues
  - Cryptocurrency-mania

Take-aways
- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.
  - Pandemic supply-chain issues
  - Cryptocurrency-mania

1. Economies of scale are a thing, not just in terms of cost. Companies dedicated to make e.g. GPUs will make a much more stable and mature product than a bunch of astronomers trying to get something to work well enough for them to do some science.
2. In no small part, anyway. Making use of stable mature products developed elsewhere applies to software tools too.
3. Despite these things, MKE CBF built in shorter time and with fewer engineers and less money than MK. Fingers crossed.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.
  - Pandemic supply-chain issues
  - Cryptocurrency-mania
- MeerKAT Extension aiming for mid-2024, so watch this space! MeerKAT has already done good science, MKE will do better.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.
  - Pandemic supply-chain issues
  - Cryptocurrency-mania
- MeerKAT Extension aiming for mid-2024, so watch this space! MeerKAT has already done good science, MKE will do better.

1. Economies of scale are a thing, not just in terms of cost. Companies dedicated to make e.g. GPUs will make a much more stable and mature product than a bunch of astronomers trying to get something to work well enough for them to do some science.
2. In no small part, anyway. Making use of stable mature products developed elsewhere applies to software tools too.
3. Despite these things, MKE CBF built in shorter time and with fewer engineers and less money than MK. Fingers crossed.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.
  - Pandemic supply-chain issues
  - Cryptocurrency-mania
- MeerKAT Extension aiming for mid-2024, so watch this space! MeerKAT has already done good science, MKE will do better.
- SKA coming to South Africa means radio astronomy will keep engineers and programmers busy for many years.

Take-aways

- Outsource hardware development where possible.
- Real-time software correlators are now a thing, thanks to Python!
- Going software-only doesn't eliminate challenges entirely.
  - Pandemic supply-chain issues
  - Cryptocurrency-mania
- MeerKAT Extension aiming for mid-2024, so watch this space! MeerKAT has already done good science, MKE will do better.
- SKA coming to South Africa means radio astronomy will keep engineers and programmers busy for many years.

science & innovation
Department:
Science and Innovation
REPUBLIC OF SOUTH AFRICA

NRF National Research Foundation
SARAO South African Radio Astronomy Observatory

Think this was cool? Get in touch, we are hiring!

James Smith
DSP Engineer
Project Lead: MeerKAT Ext CBF
Email: jsmith@sarao.ac.za

www.sarao.ac.za

The South African Radio Astronomy Observatory (SARAO) is a National Facility managed by the National Research Foundation and incorporates all national radio astronomy telescopes and programmes.

CAPE TOWN Tel: +27 (0)21 506 7300 | 2 Fir Street, Black River Park | Observatory, Cape Town | South Africa 7925
HARTEBEESTHOEK Tel: +27 (12) 301 3100 | Farm 502 JQ, Hartebeesthoek, Broederstroom Road, Hartebeesthoek, 1740

Think this was cool? Get in touch, we are hiring!
James Smith
DSP Engineer
Project Lead: MeerKAT Ext CBF
Email: jsmith@sarao.ac.za