An Initial Framework for Prototyping Radio-Interferometric Imaging Pipelines

Sunrise Wang, Nicolas Gac, Hugo Miomandre, Jean-Francois Nezan,
Karol Desnos, Francois Orieux

Introduction

Prototype framework to estimate resource usage of Radio-Interferometric imaging pipelines

Aimed at large radio-telescopes e.g. SKA

- Large amounts of data
- conflicting design restrictions

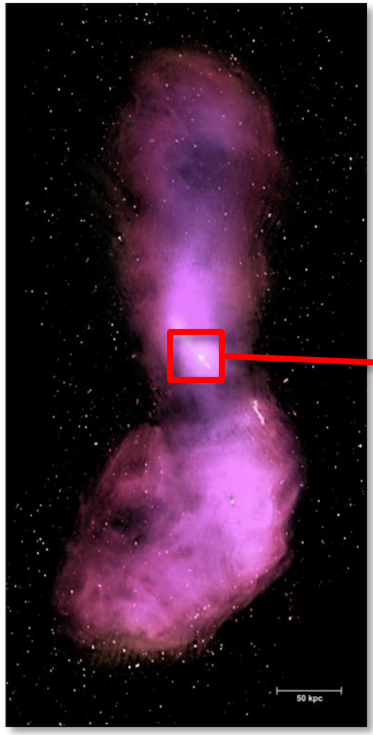
The square kilometer array in south africa (left) and australia (right)[1]

Aid in designing super-computer hardware and software architecture

[1]https://www.skao.int/

Radio-Interferometry

Seeing the universe in radio



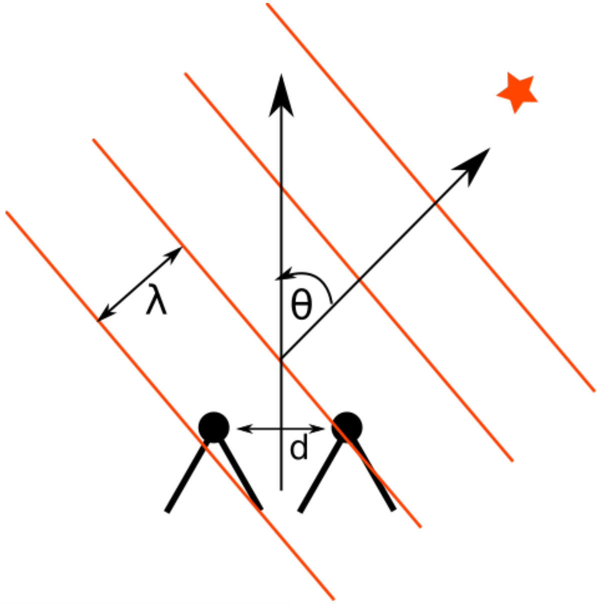
Centaurus A @ ~1.4Ghz (z=~0.0018) [1]

Centaurus A in visible spectrum[2]

Antenna arrays + Radio-Interferometry allows better angular resolution and sensitivity compared to single dishes

[1]Feain, Ilana & Cornwell et al. (2011). ASTROPHYS J. 740. 10.1088/0004-637X/740/1/17. [2]https://www.eso.org/public/france/images/eso0315a/

Sampling the Sky



Sample by correlating antenna pairs ie. baselines

Each sample (ie. visibility) can expressed as:

True sky

$$V(u,v,w) = \int\int \frac{I(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i[ul+vm+w(\sqrt{1-l^2-m^2}-1)]} \mathrm{d}l\mathrm{d}m$$

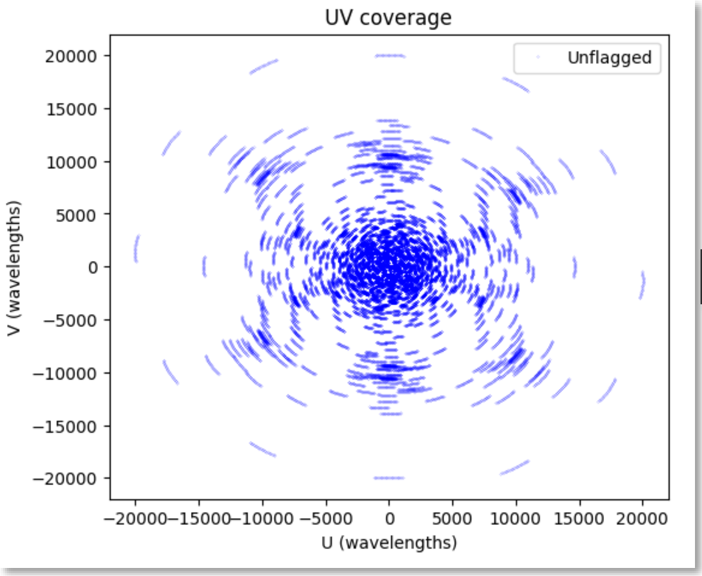
Difference in antenna positions in earth's rotation frame

Non-coplanar baseline

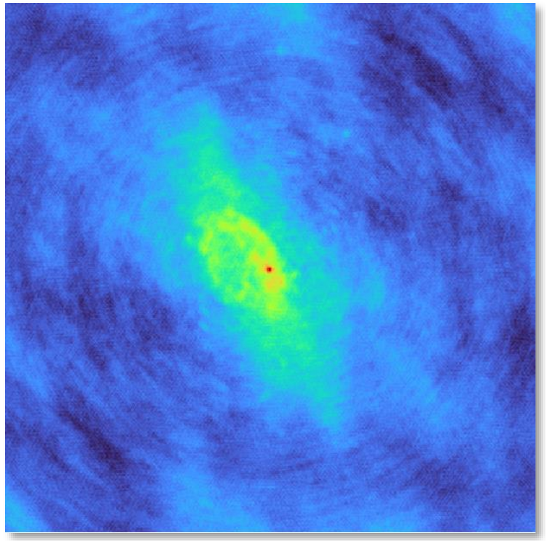
Spatial (angular) coordinates

Imaging

36 antennas
(ASKAP)



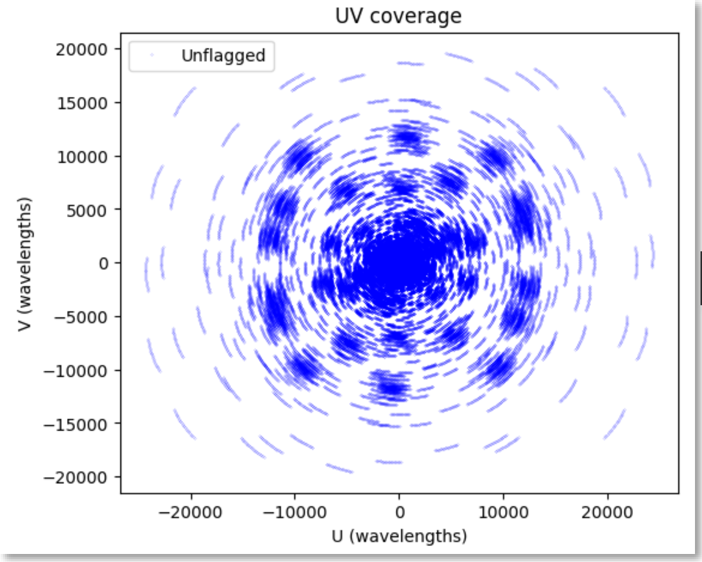
iFT



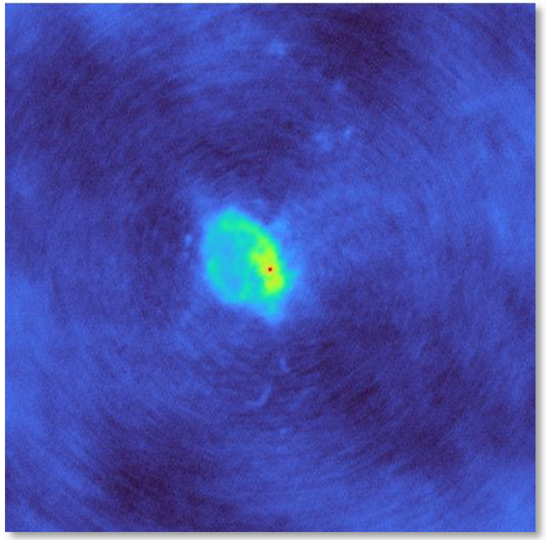
Imaging pipeline corrects for artefacts

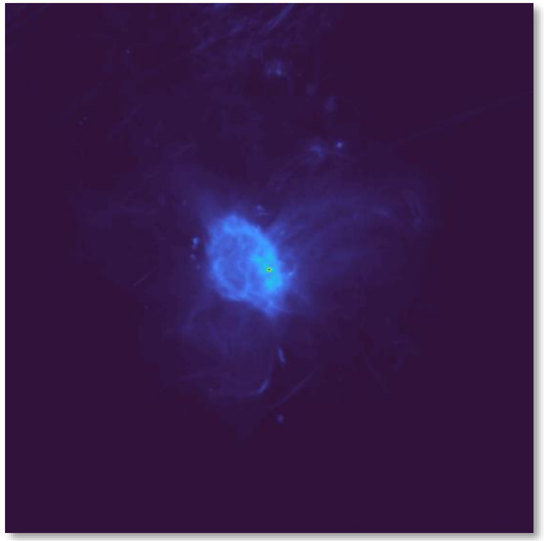
- Limited by number of samples and sensitivity

64 antennas
(MEERKAT)



iFT





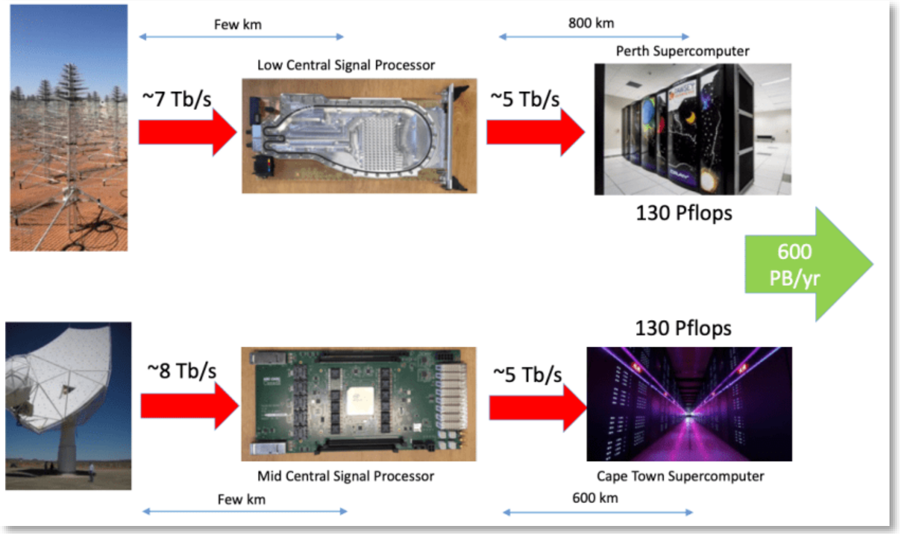
True sky, Sgr A[1]

[1]Heywood, I., et al. "The 1.28 ghz meerkat galactic center mosaic." *The Astrophysical Journal* 925.2 (2022): 165.

Increasing the number of antennas

Increases amount of data quadratically!

$$n_{vis} = \frac{n_{ant}(n_{ant}+1)}{2}$$



Antennas to SDP [1]

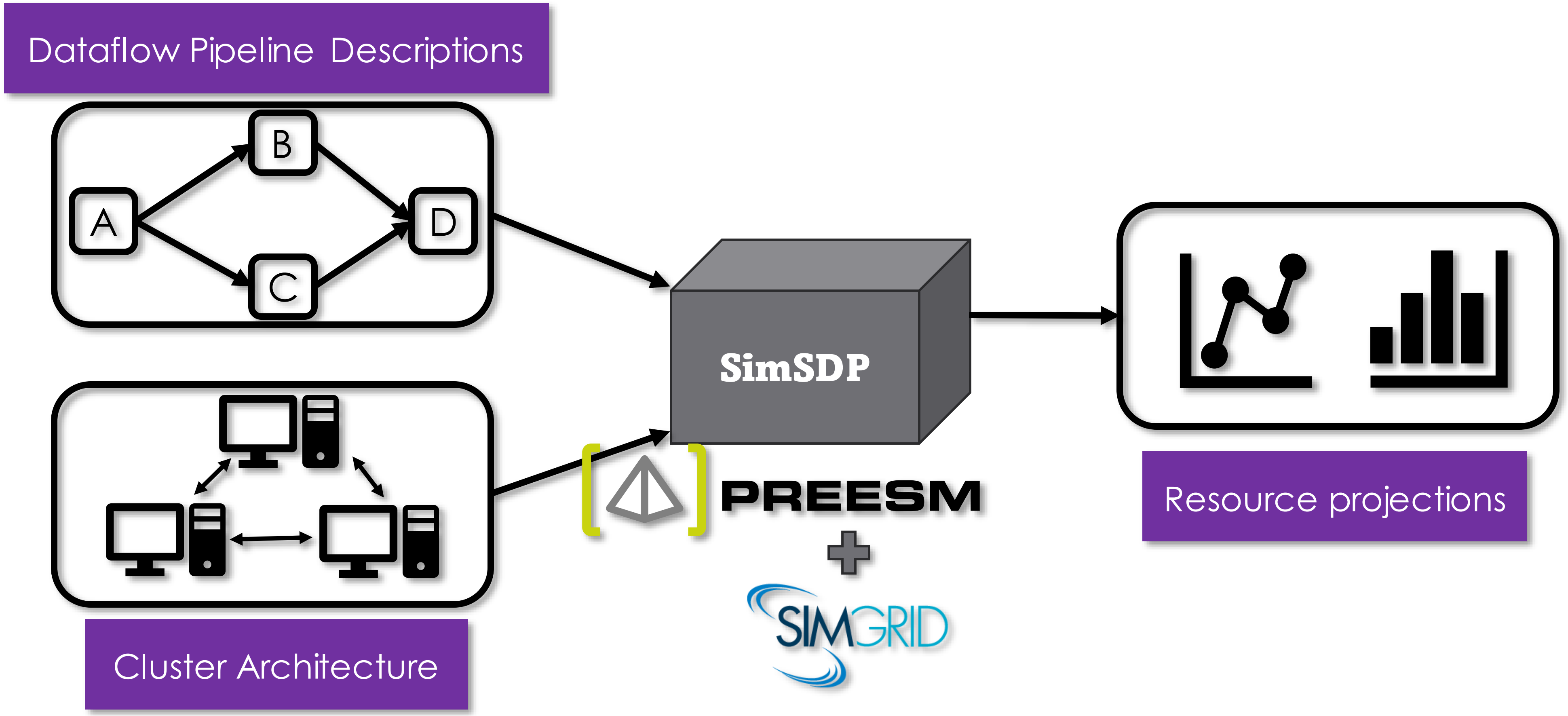
Science Data Processor ingest stream projected to be around 0.4 TB/s ≈ 34.5 PB/day

- Storage expensive, hard time constraints
- Introduces a lot of data-transfer overhead
- Energy costs
- Different pipelines for different types of science (e.g. continuum vs spectral line)

[1]https://research.csiro.au/ska/technology/

SimSDP

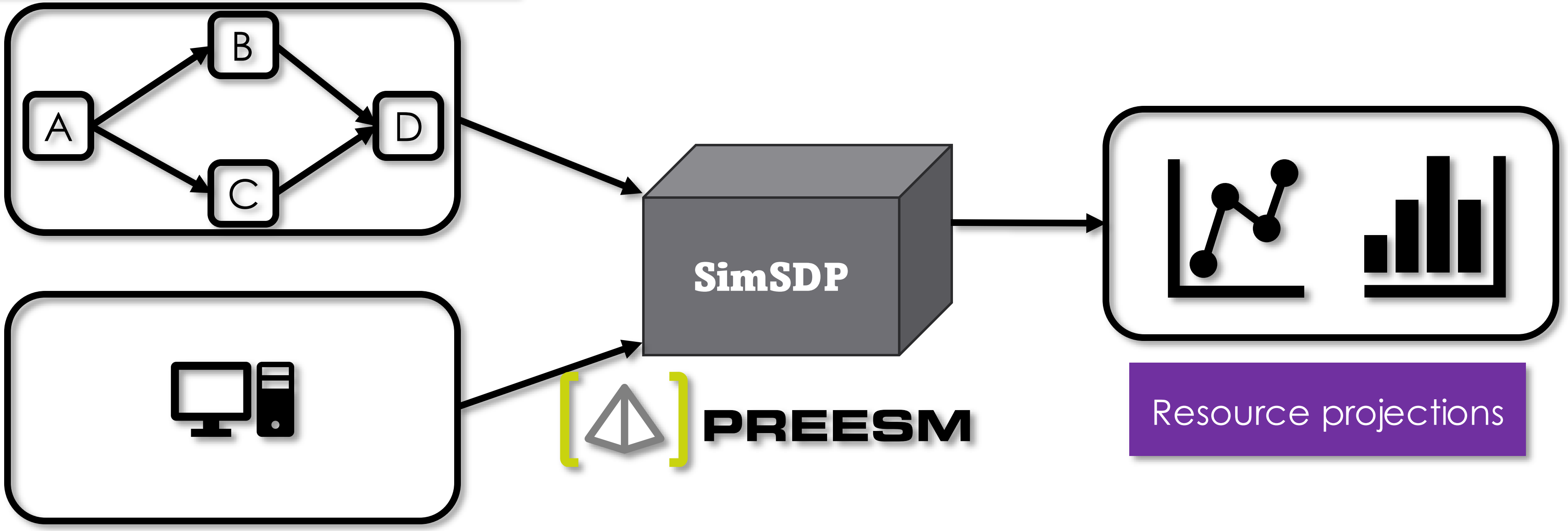
Aid in design of software and hardware architecture by simulating resource usage



SimSDP

Aid in design of software and hardware architecture by simulating resource usage

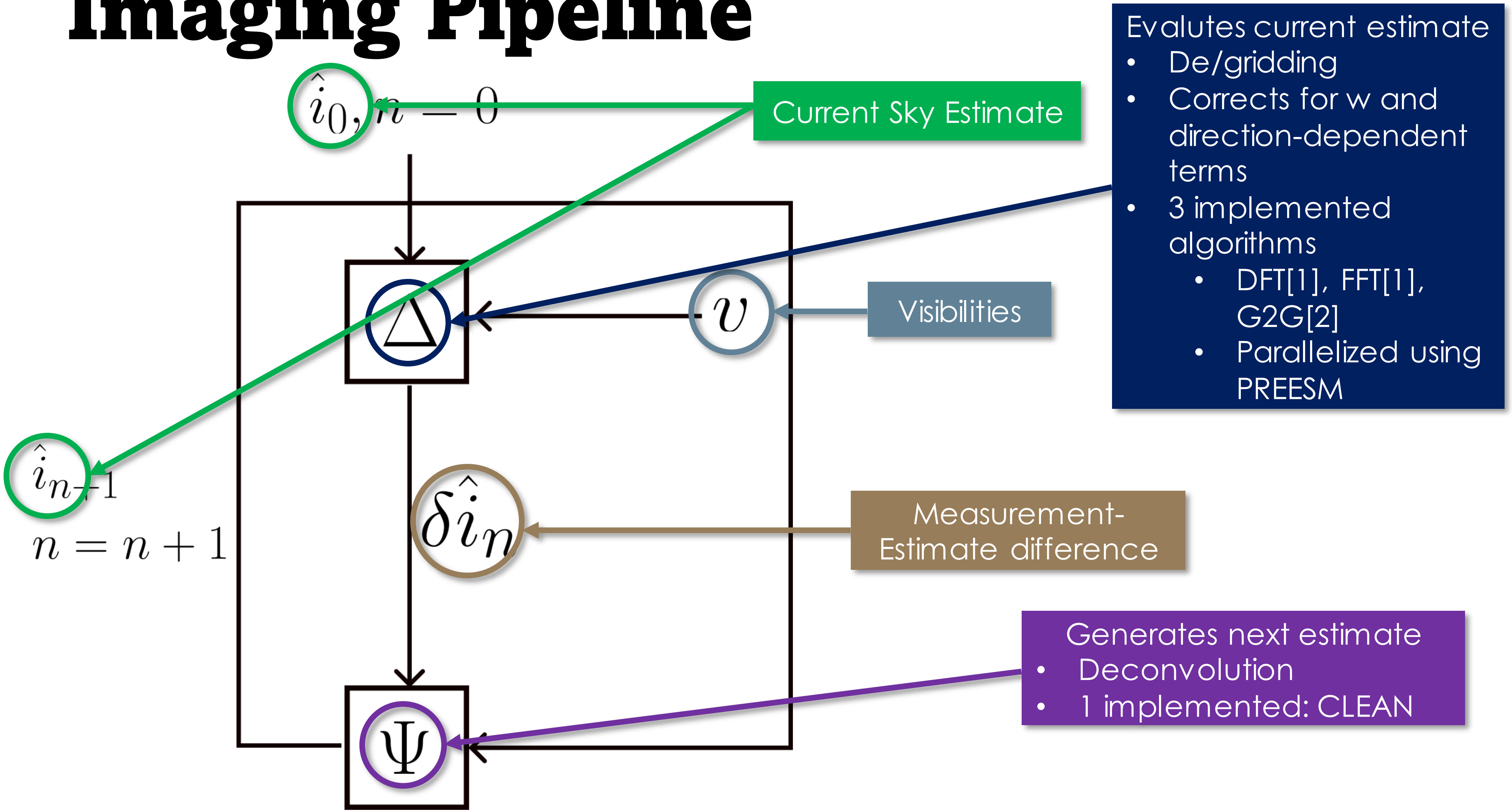
Dataflow Pipeline Descriptions



Resource projections

Local machine Architecture

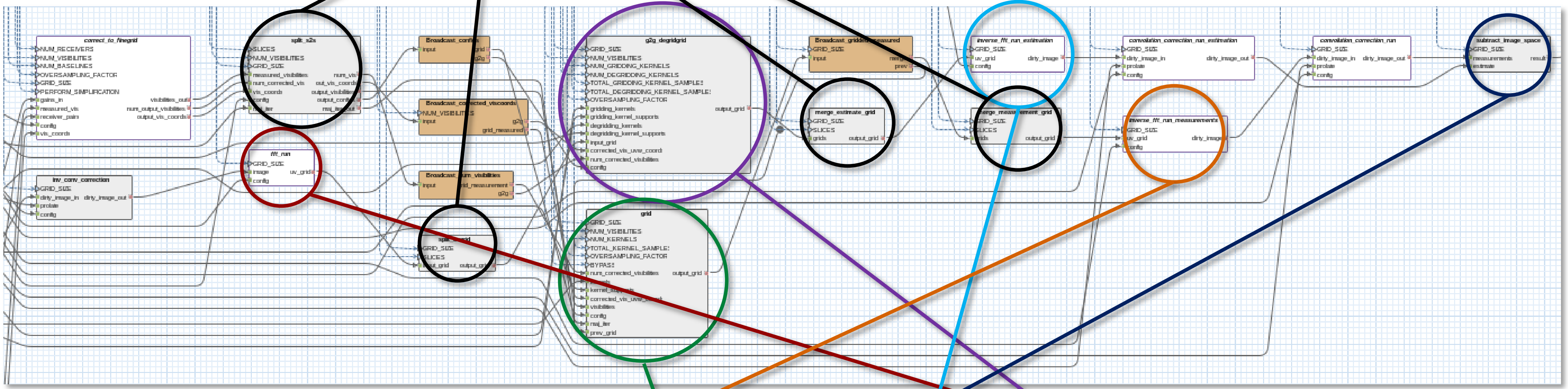
The Radio-Interferometric Imaging Pipeline



[1][2]Schwab, F. R. (1984b), ``Relaxing the isoplanatism assumption in self-calibration; applications to low-frequency radio interferometry", Astron. J., 89, 1076-1081.
[2]Monnier, Nicolas, et al. "Fast grid to grid interpolation for radio interferometric imaging." Astronomy and Computing 45 (2023): 100767.

Concrete Example: Grid 2 Grid

Split/Merge for automatic parallelism

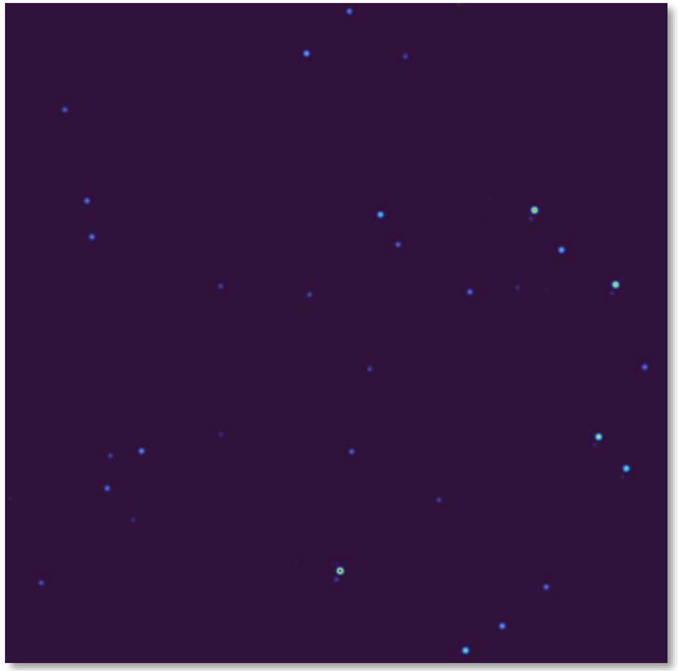


$$\delta\hat{i}_n = F^\dagger G^\dagger v - F^\dagger G^\dagger G F \hat{i}_n$$

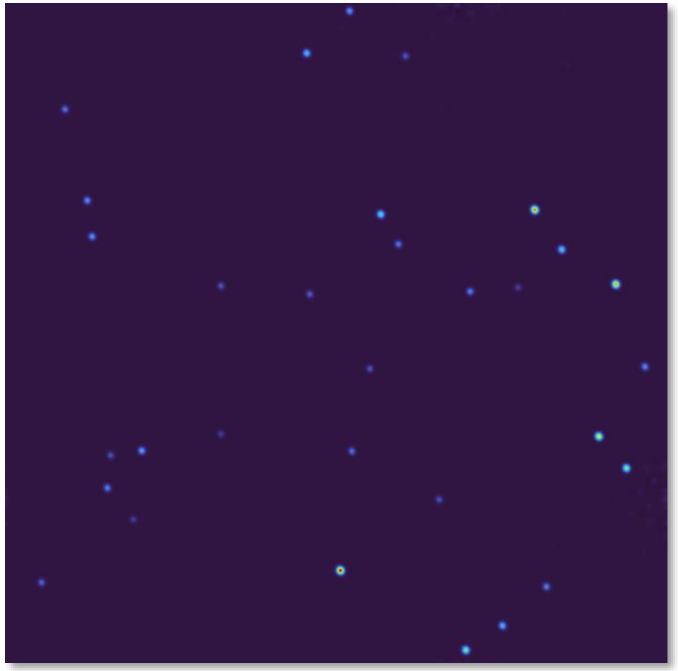
Evaluation

Implement actors as C functions, generate entire pipeline code with PREESM.

Compare output against another imaging system (RASCIL[1])



Ours



RASCIL

Compared measured against estimated

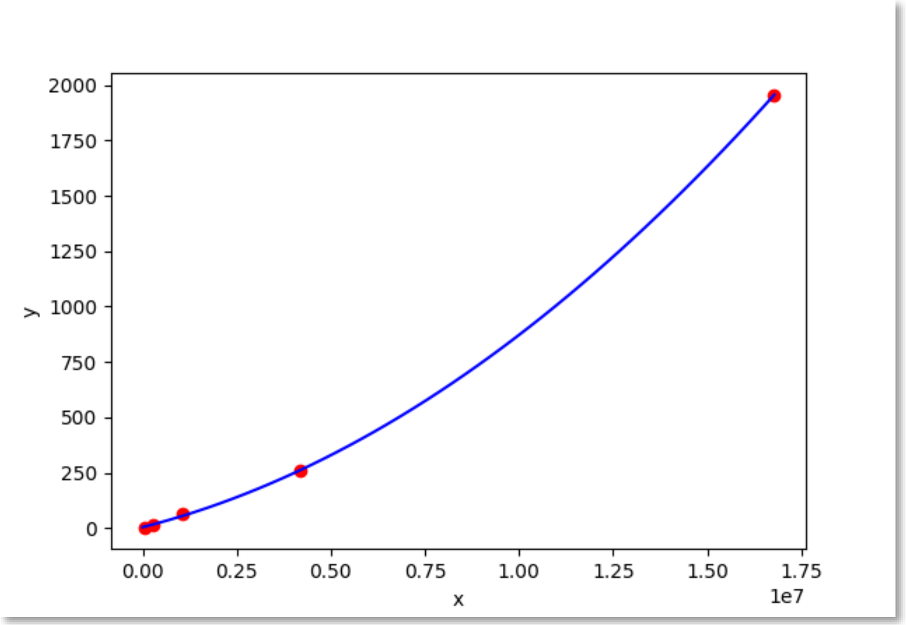
- Memory
- Computation Time

[1] T. J. Cornwell et al., Radio astronomy simulation, calibration and imaging library, 2020

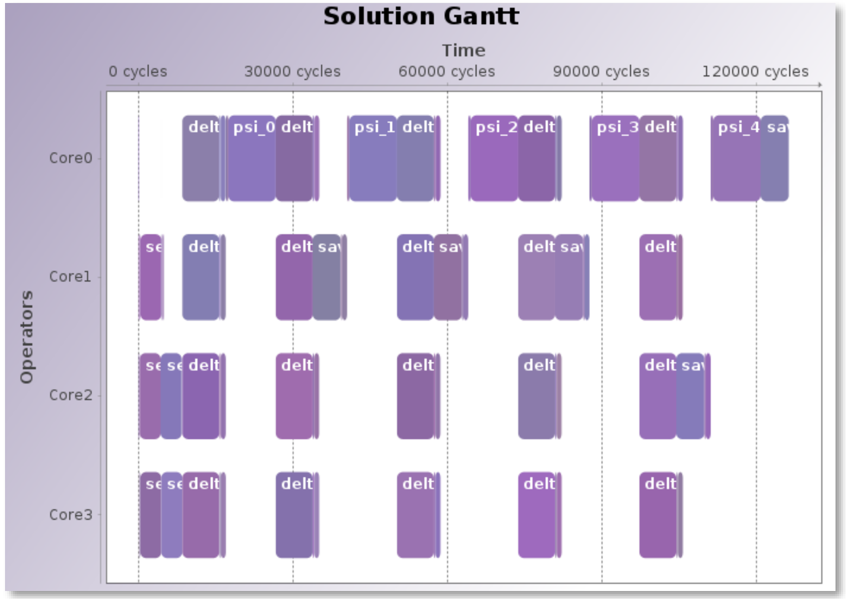
Estimating Computation Time

Ran benchmarks for each actor, either with our or with optimized code

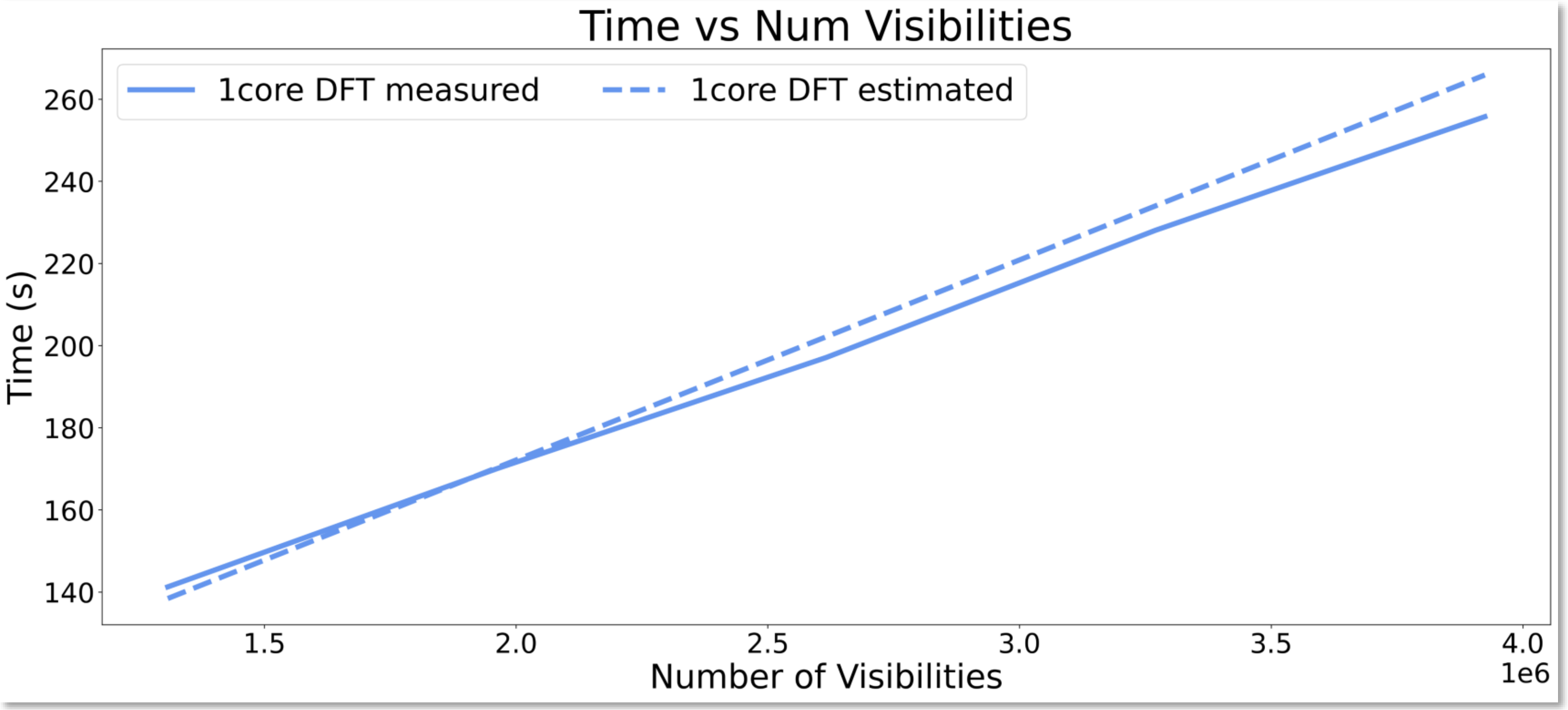
- Varied number of visibilities, grid-size, and number of minor cycles
- Fitted polynomial of appropriate degree



Estimated time obtained via PREESM gantt chart

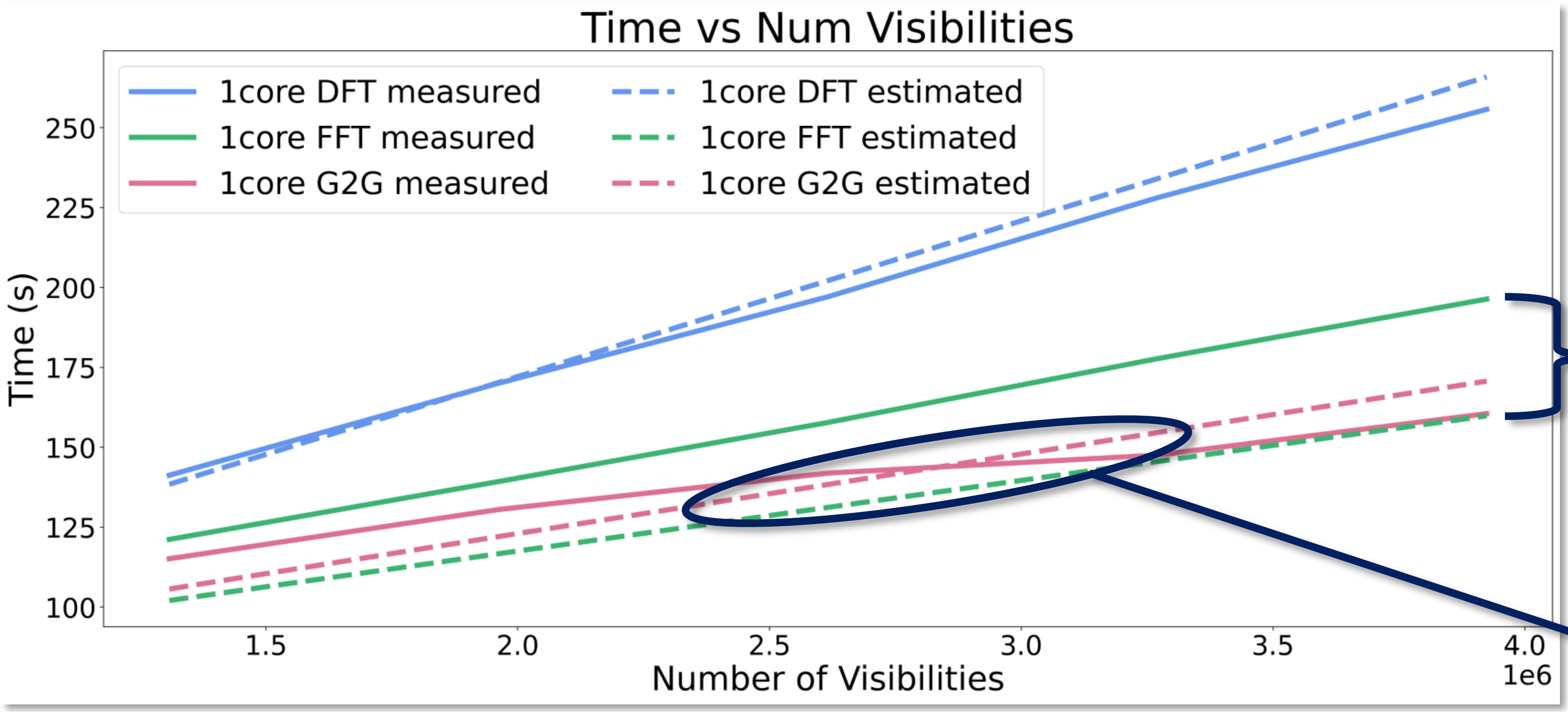


Estimated vs Measured Computation times



Estimation mostly correctly, difference primarily due to error in fitting model

Estimated vs Measured Computation times

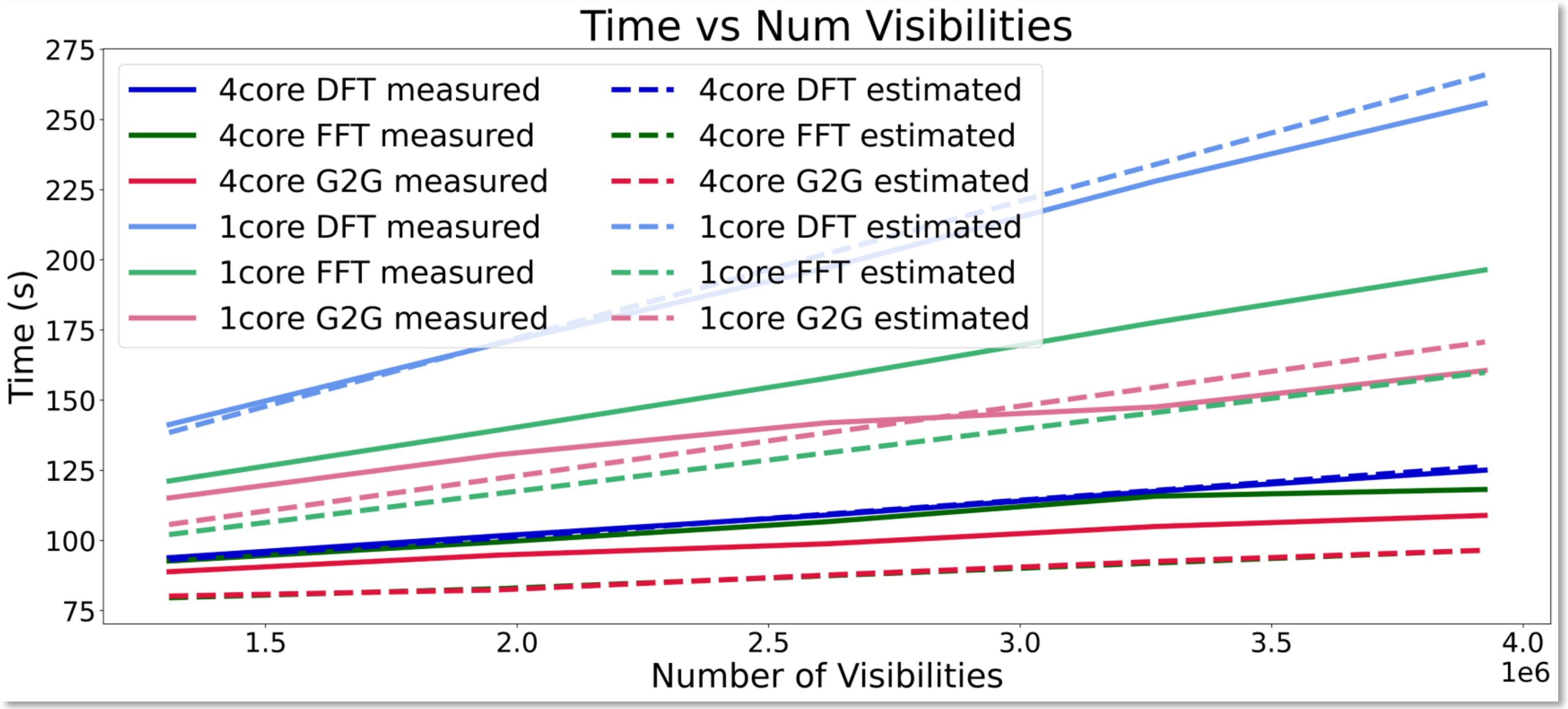


Trend still mostly correct

Benchmark different from version run

G2G employs data compression, static analysis does not account for this

Estimated vs Measured Computation times



Predicts correctly the 1.5-3x speedup when parallelizing

Estimated vs Measured Computation times

Framework predicts well:

- General algorithmic complexity
- Performance gain from parallelization

Estimation limitations:

- Benchmark different to measured
- Fitting error
- Only static analysis

Resource Estimation – Max Memory

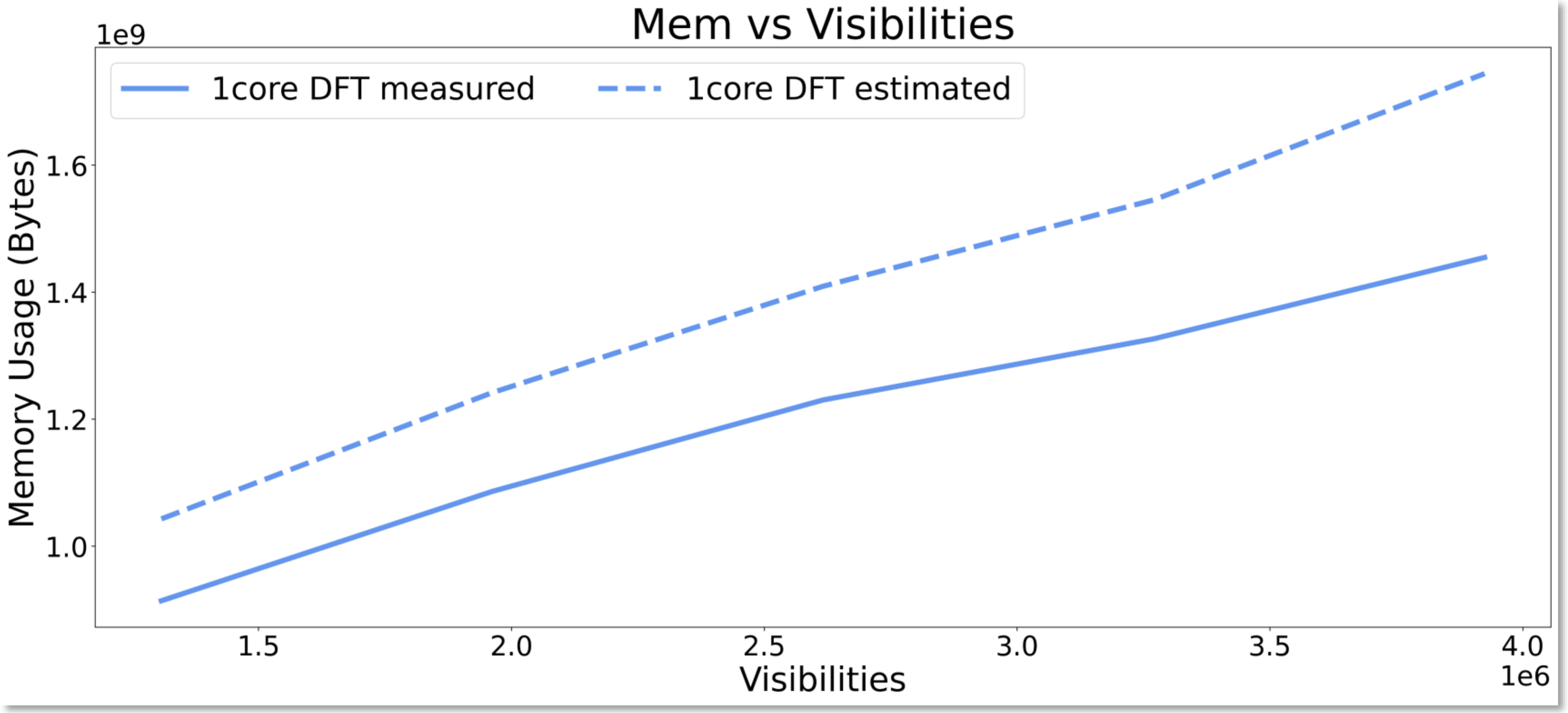
Use PREESM's reported allocated inter-node memory as estimation

```
Starting allocation with BestFitAllocator(LARGEST_FIRST)
BestFitAllocator(LARGEST_FIRST) allocates 3.2940004020929337 GBytes in 171 ms.
```

Use GNU time tool's reported maximum resident set size for measured

```
Average total size (kbytes): 0
Maximum resident set size (kbytes): 914240
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 56
```

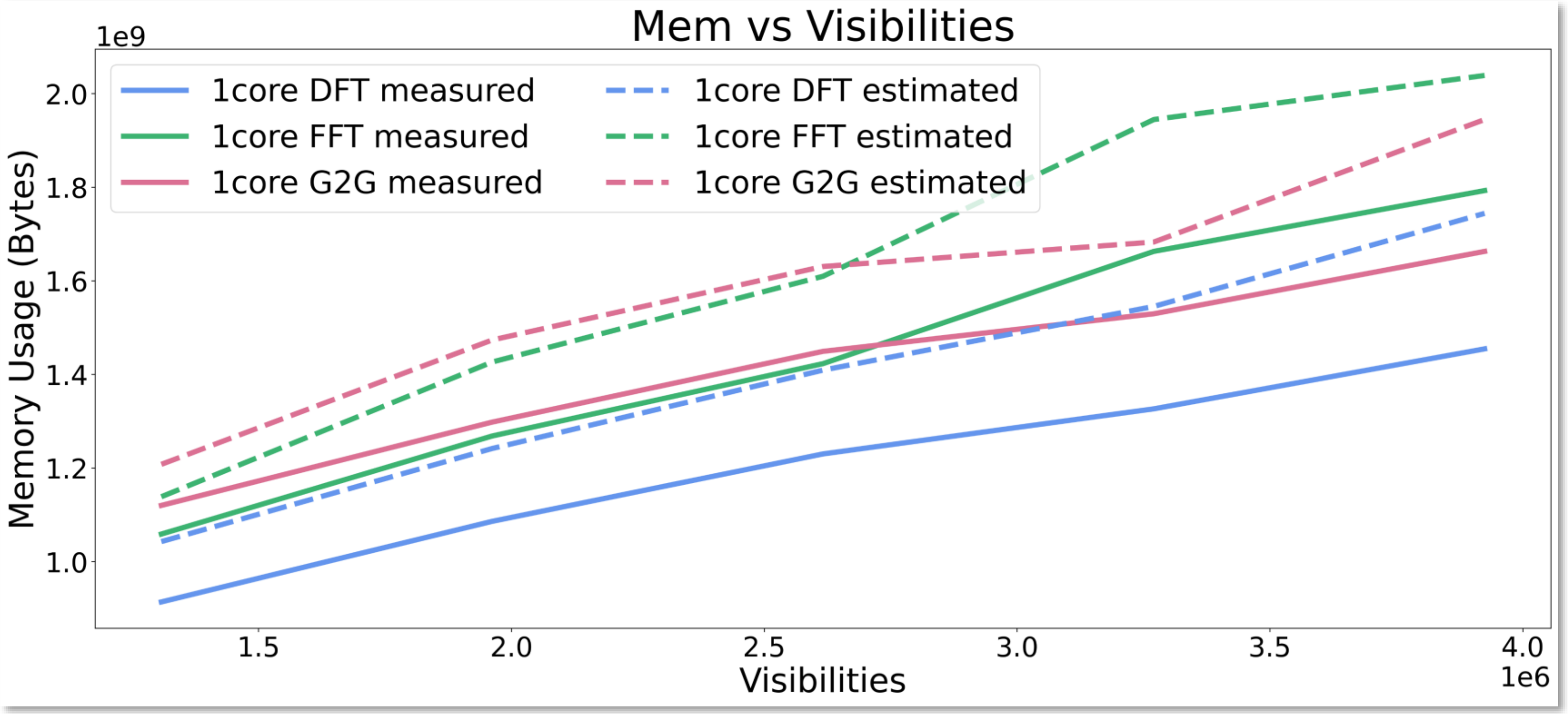
Resource Estimation – Memory: Results



Predicts the general trend.

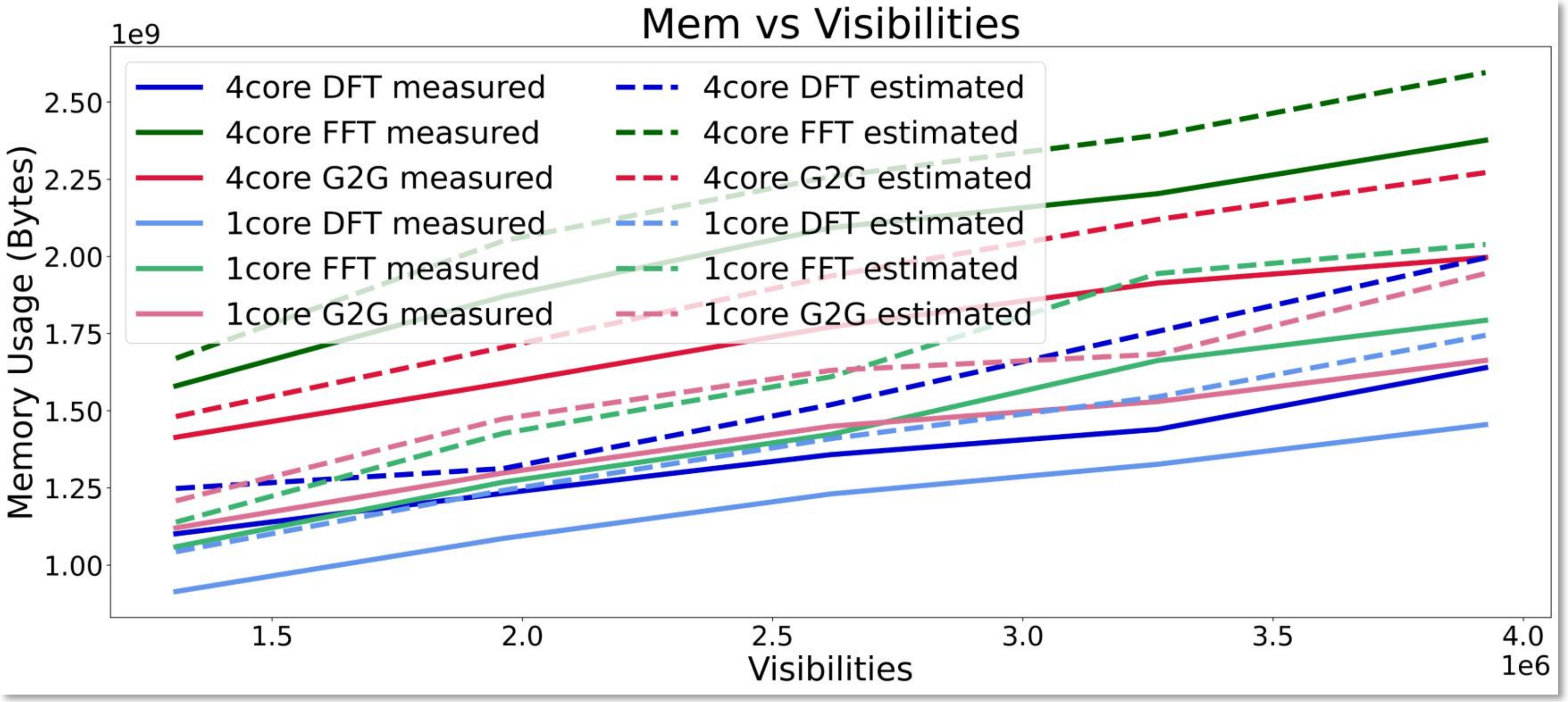
Unexpected that measured < estimated (should be the inverse). Need to investigate

Resource Estimation – Memory: Results



Similar results for FFT and G2G pipelines

Resource Estimation – Memory: Results



Accurately predicts 1.1-1.5x increase in memory when increasing parallelization

Resource Estimation – Memory: Conclusions

Framework does a good job in predicting memory increase both as algorithm parameters, and parallelization increases

Measured always a bit less than estimated. Should not be the case and need to test with valgrind which should give a more thorough profiling

Conclusions

Introduced an initial framework to prototype radio-interferometric algorithms

- Estimates computation time and memory
- Promising results, estimations similar trend to measured

Aspects to improve

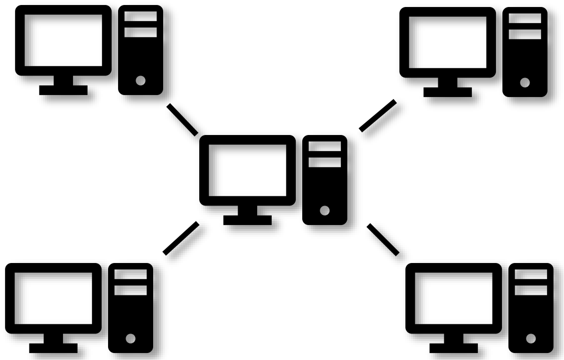
- Improve models and benchmark data for better estimations
- Optimize implemented pipelines

Main drawback is that only static analysis is supported currently

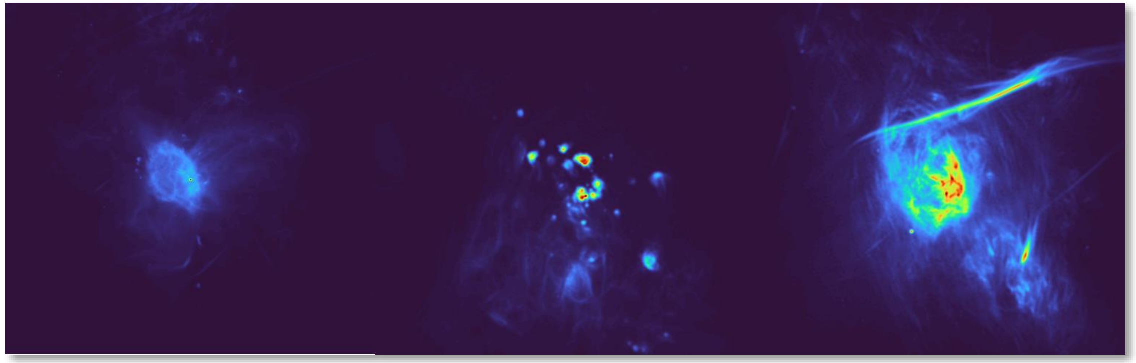
- Runtime/data dependent aspects e.g. compression can change runtime
- SPIDER can solve this

Future work

Integrate support for different hardware architectures, evaluate on clusters, and in cases where we cannot obtain measurements (e.g. massive datasets)



Add support for more pipelines and datasets



Questions?