



HAL
open science

An Initial Framework for Prototyping the Radio-Interferometric Imaging Pipelines

Sunrise Wang, Nicolas Gac, Hugo Miomandre, Jean-François Nezan, Karol
Desnos, François Orioux

► **To cite this version:**

Sunrise Wang, Nicolas Gac, Hugo Miomandre, Jean-François Nezan, Karol Desnos, et al.. An Initial Framework for Prototyping the Radio-Interferometric Imaging Pipelines. DASIP 2024 - Workshop on Design and Architectures for Signal and Image Processing, HiPEAC, Jan 2024, Munich, Germany. hal-04361151v2

HAL Id: hal-04361151

<https://hal.science/hal-04361151v2>

Submitted on 11 Jan 2024 (v2), last revised 7 Feb 2024 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Initial Framework for Prototyping Radio- Interferometric Imaging Pipelines

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



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

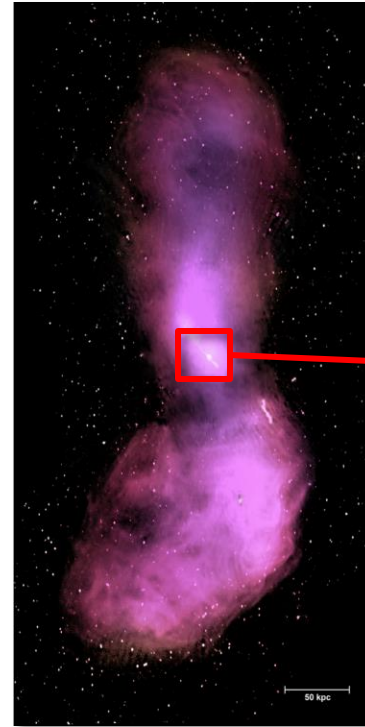


The square kilometer array in south africa (left) and australia (right)[1]

Aid in designing super-computer hardware and software architecture

Radio-Interferometry

Seeing the universe in radio



Centaurus A @ ~1.4Ghz ($z \sim 0.0018$) [1]

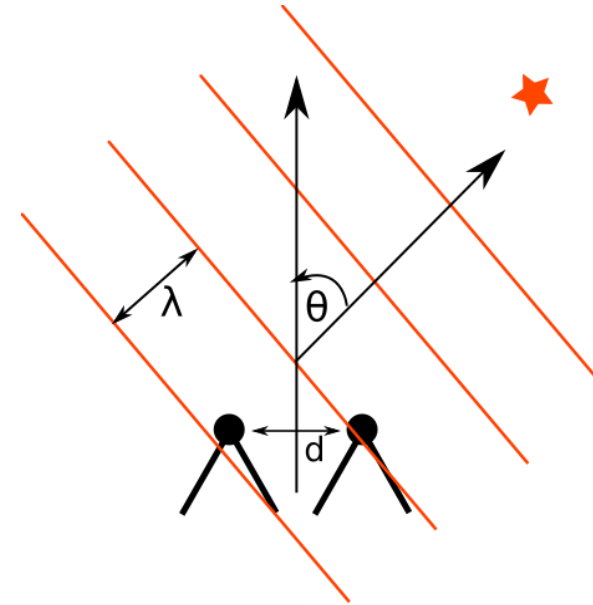


Centaurus A in visible spectrum [2]

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

Sampling the Sky

Sample by correlating antenna pairs ie. baselines



Each sample (ie. visibility) can expressed as:

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

True sky

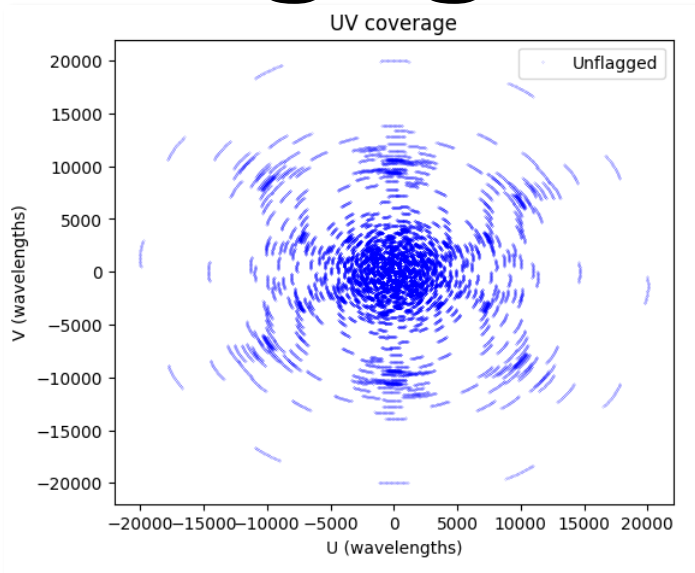
Difference in antenna positions in earth's rotation frame

Non-coplanar baseline

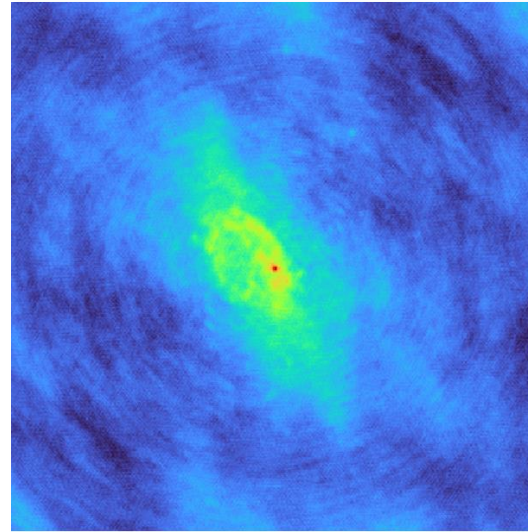
Spatial (angular) coordinates

Imaging

36 antennas
(ASKAP)



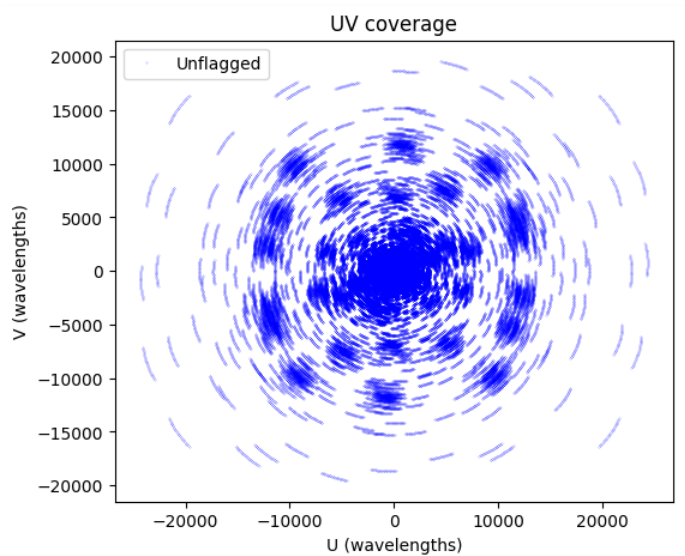
iFT



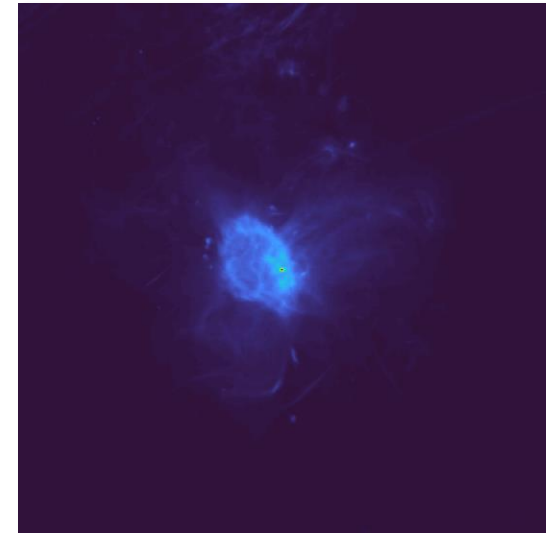
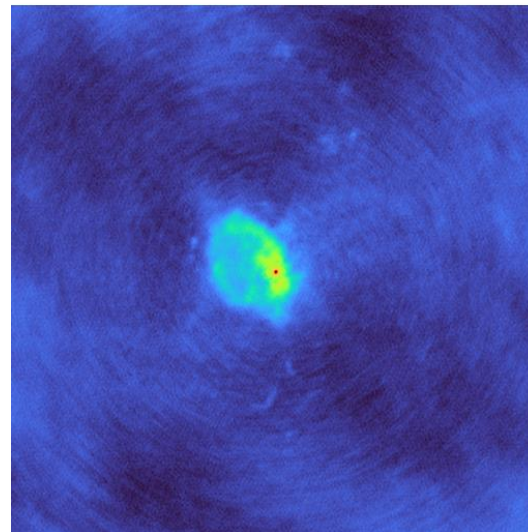
Imaging pipeline corrects for artefacts

- Limited by number of samples and sensitivity

64 antennas
(MEERKAT)



iFT

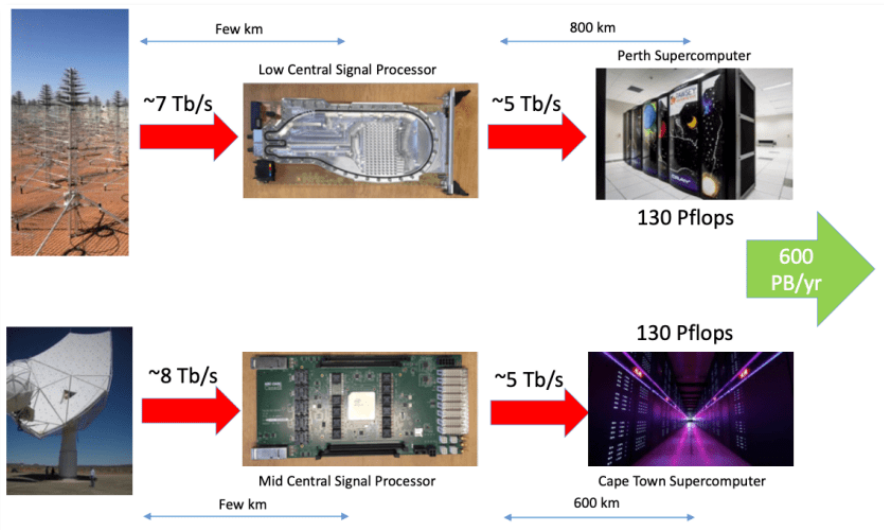


True sky, Sgr A[1]

Increasing the number of antennas

Increases amount of data quadratically!

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



Antennas to SDP[1]

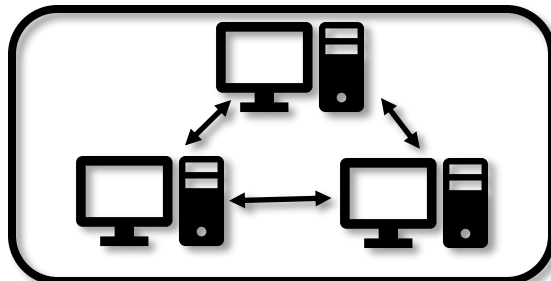
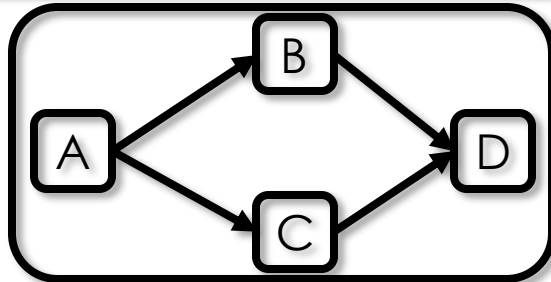
Science Data Processor ingest stream projected to be around 0.4 TB/s \approx 34.5 PB/day

- 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)

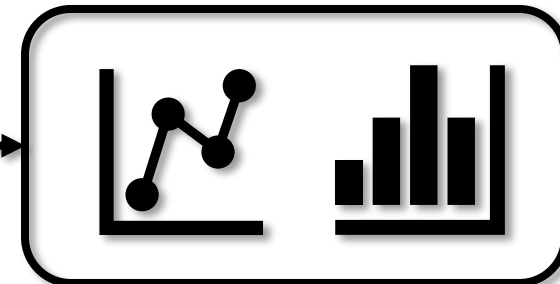
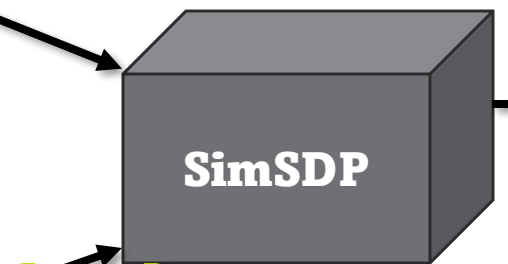
SimSDP

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

Dataflow Pipeline Descriptions



Cluster Architecture

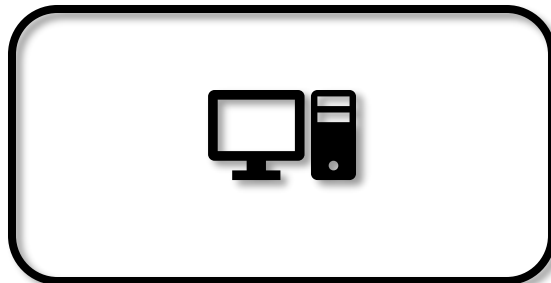
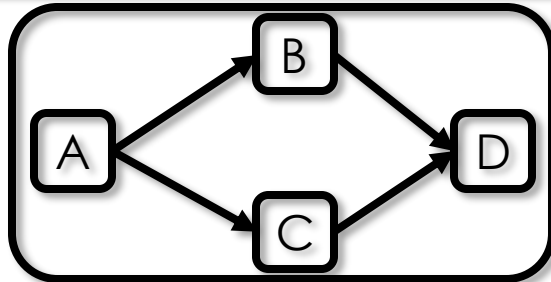


Resource projections

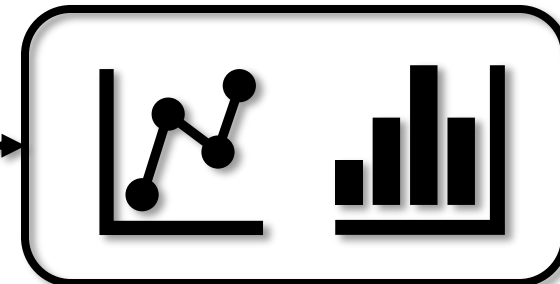
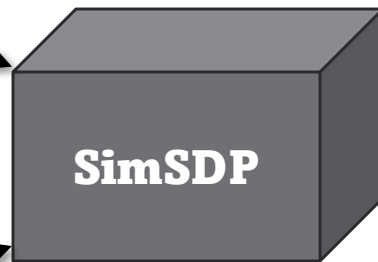
SimSDP

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

Dataflow Pipeline Descriptions

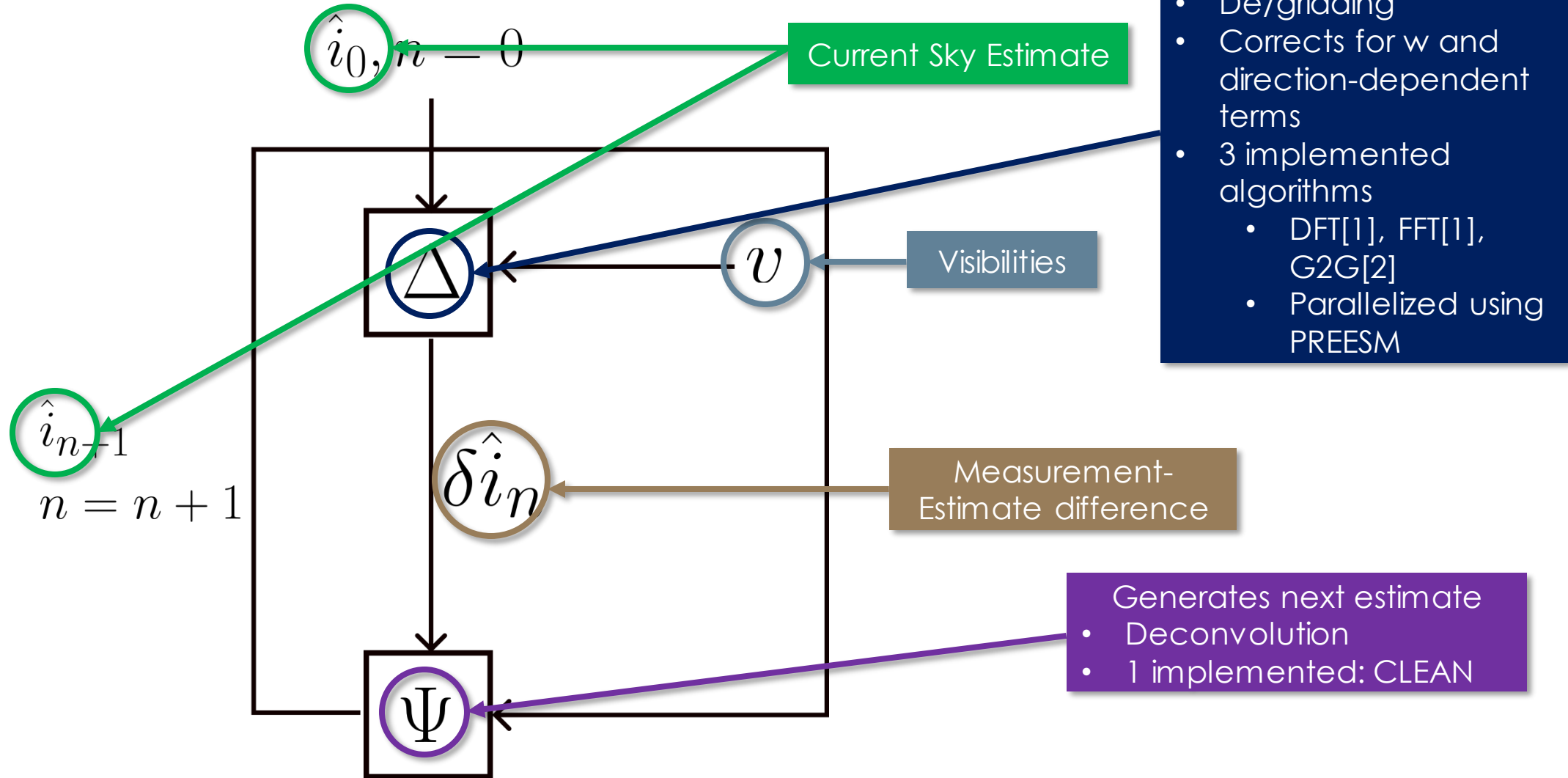


Local
machine Architecture



Resource projections

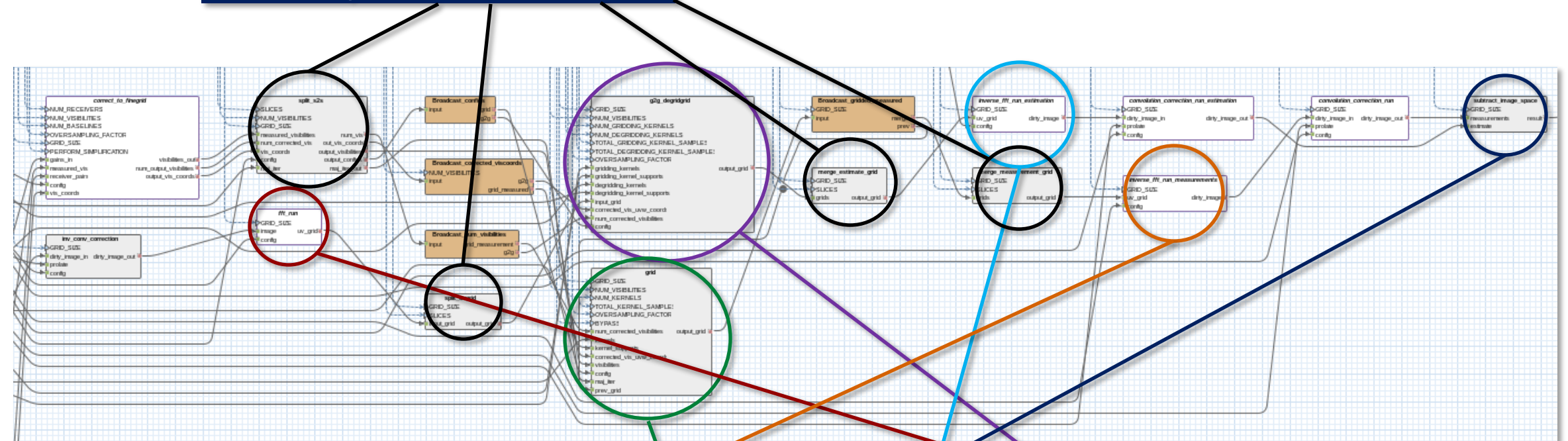
The Radio-Interferometric Imaging Pipeline



[1] Schwab, F. R. (1984b), "Relaxing the isoplanatism assumption in self-calibration; applications to low-frequency radio interferometry", *Astron. J.*, 89, 1076-1081.
 [2] Momnier, 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

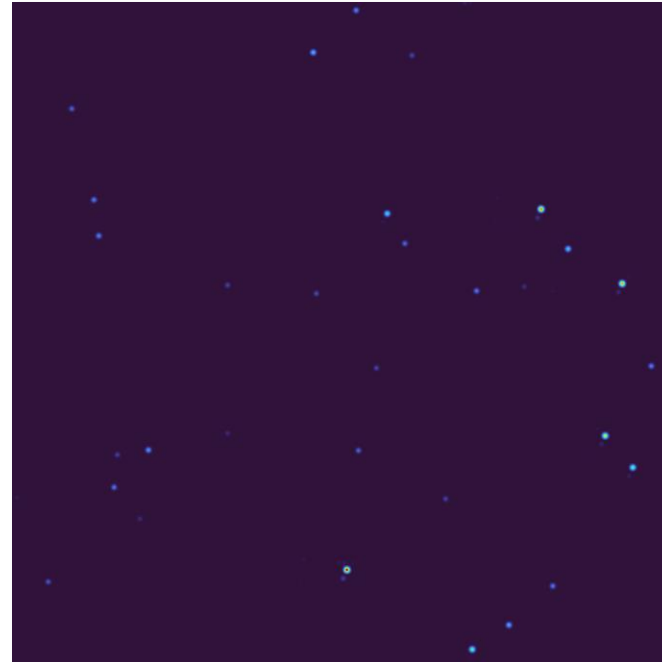


$$\hat{\delta i}_n = F^\dagger G^\dagger v - F^\dagger G^\dagger G F \hat{i}_n$$

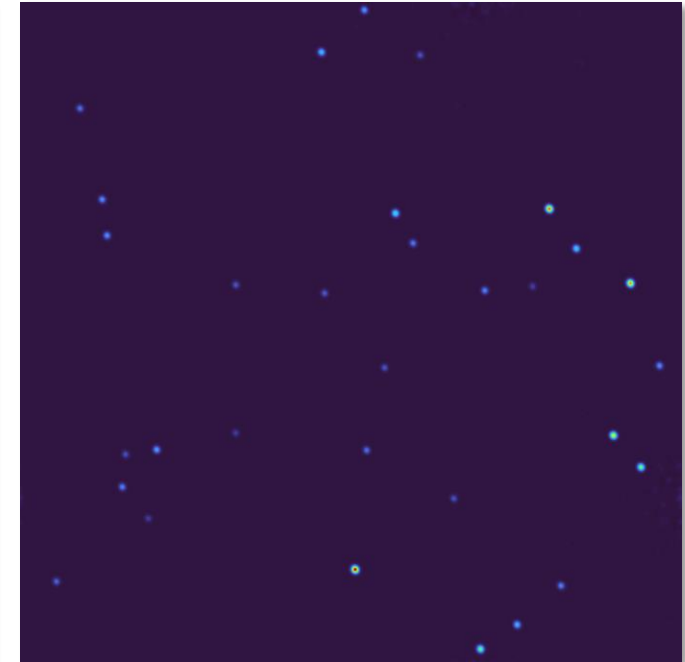
Evaluation

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

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



Ours



RASCIL

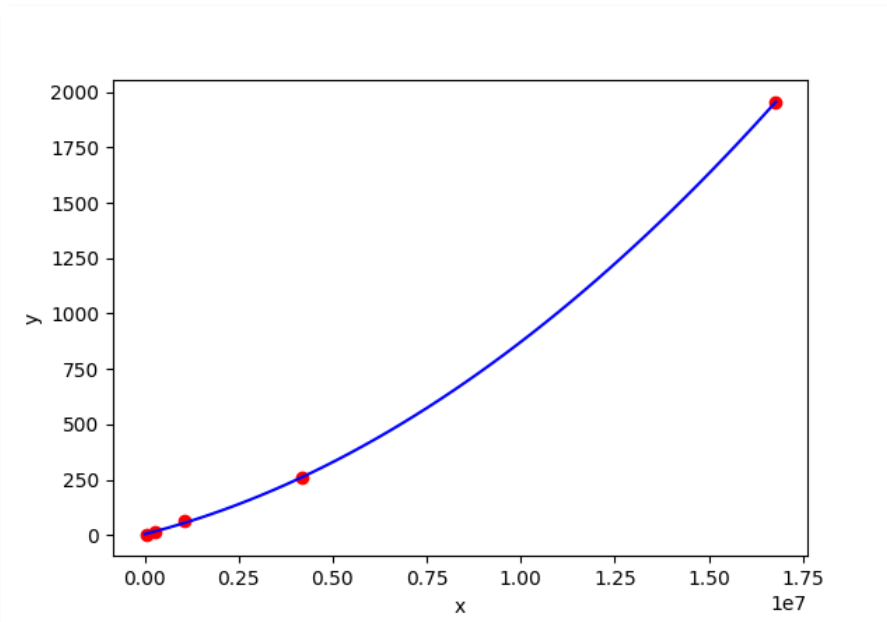
Compared measured against estimated

- Memory
- Computation Time

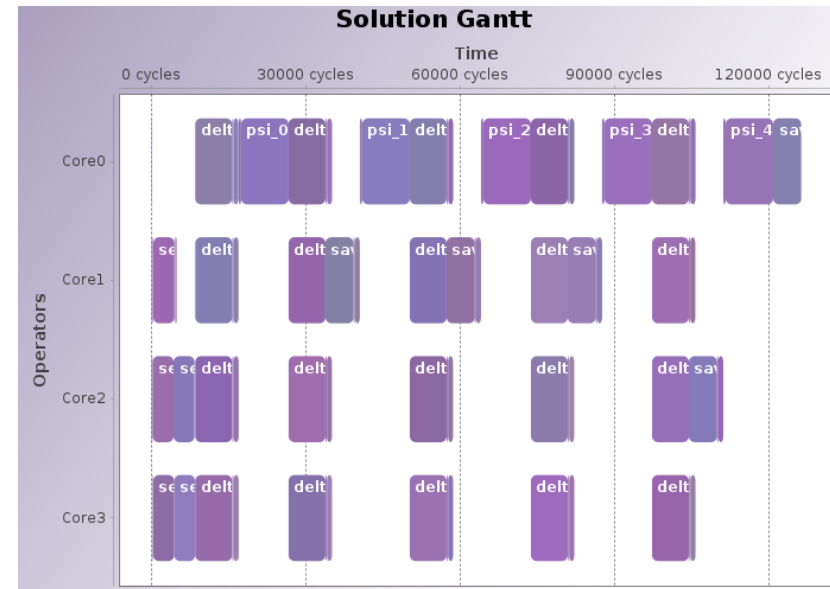
Estimating Computation Time

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

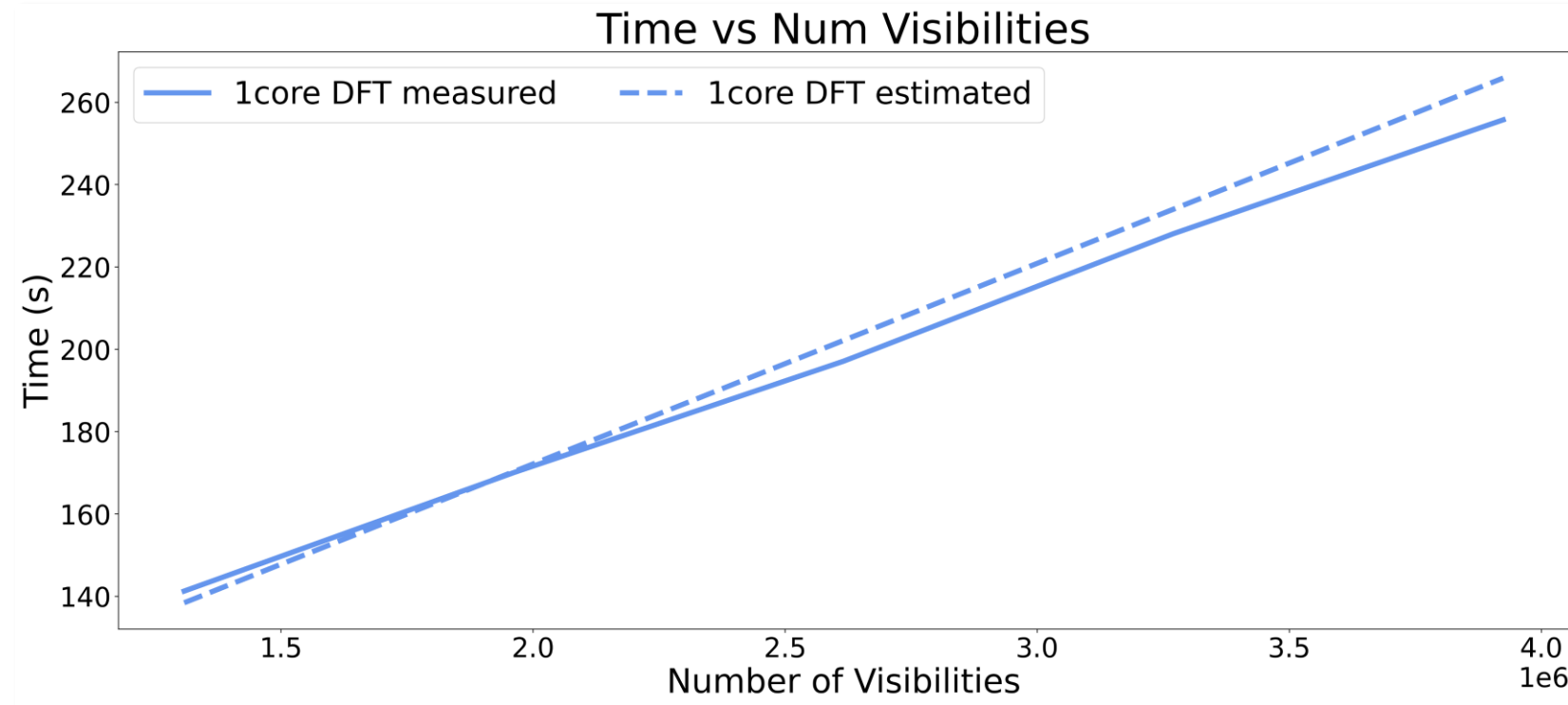
- Varied number of visibilities, grid-size, and number of minor cycles
- Fitted polynomial of appropriate degree



Estimated time obtained via PREESM gantt chart

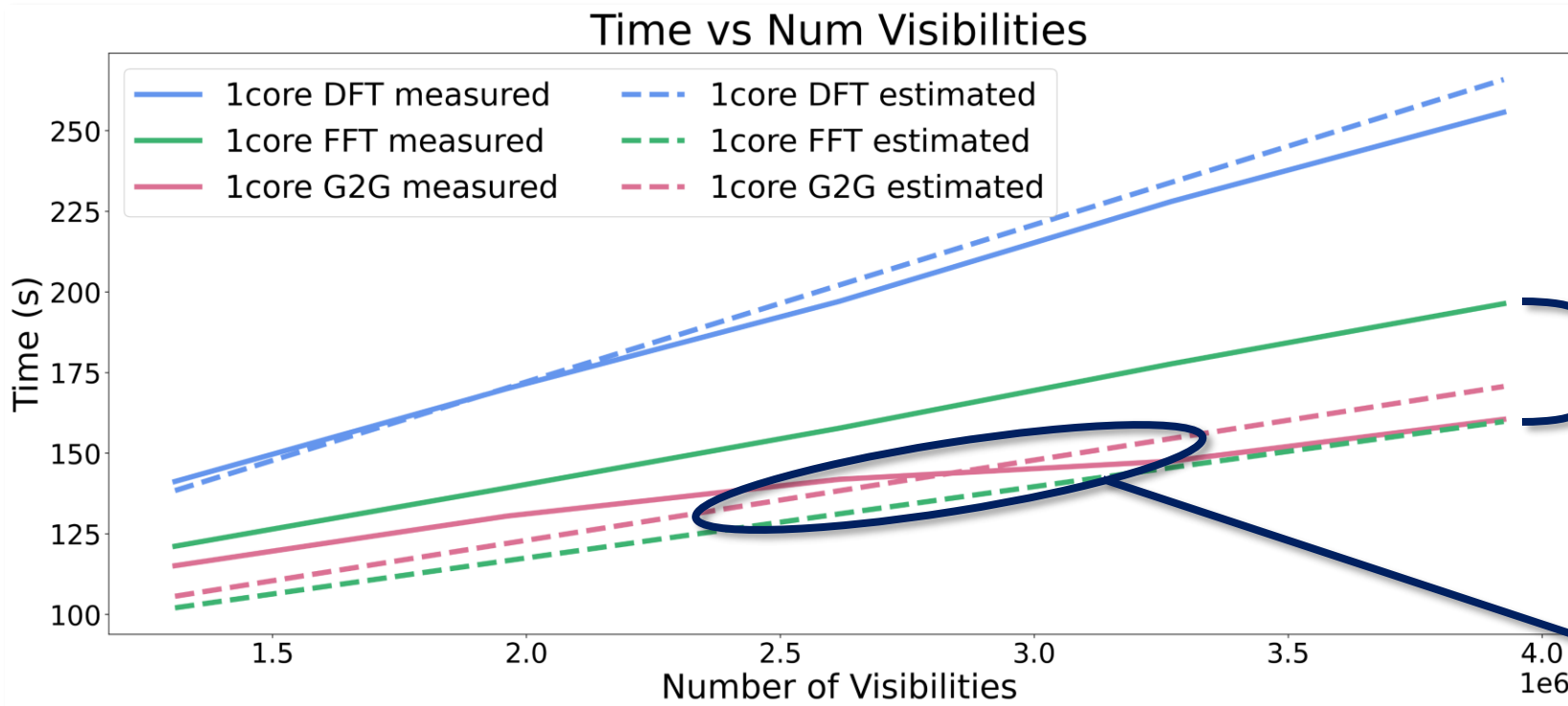


Estimated vs Measured Computation times



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

Estimated vs Measured Computation times

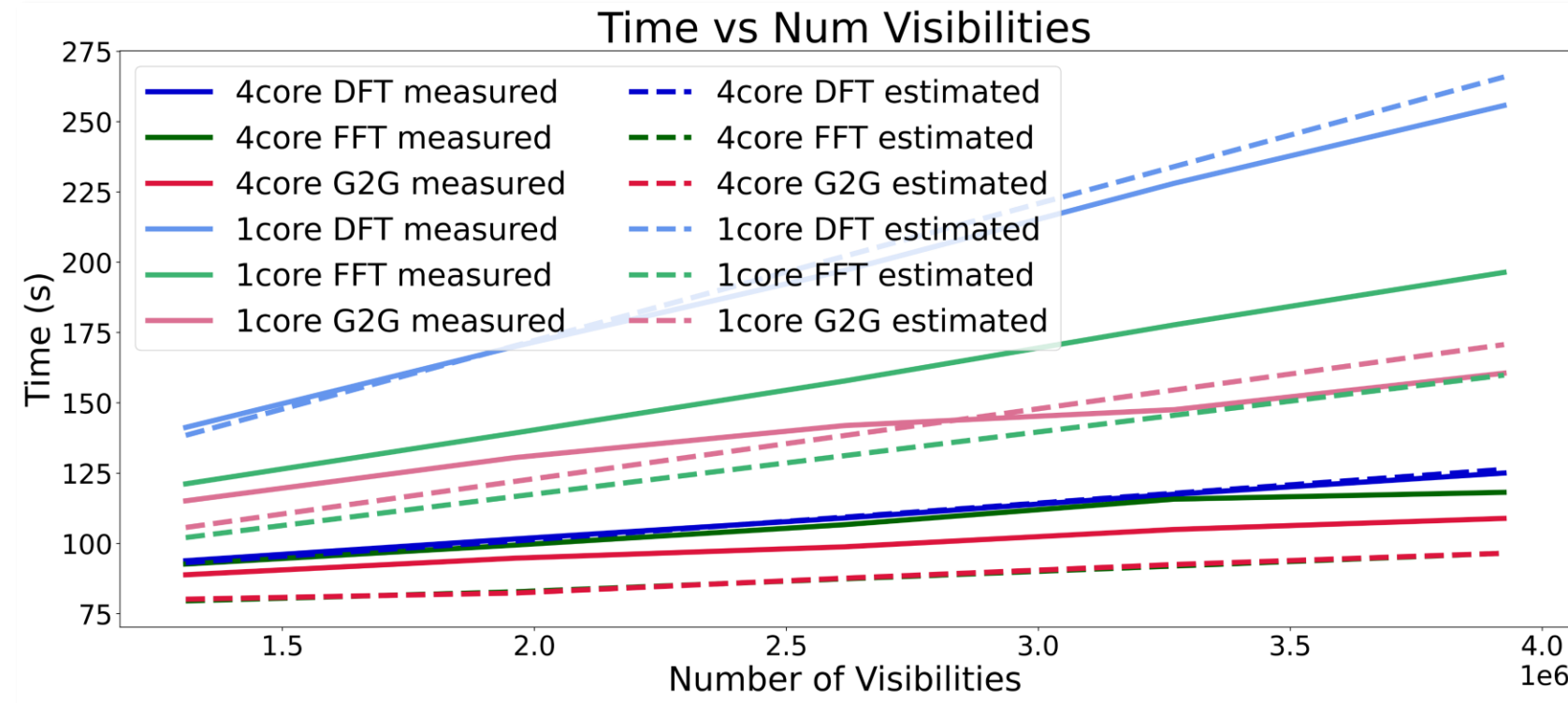


Trend still mostly correct

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

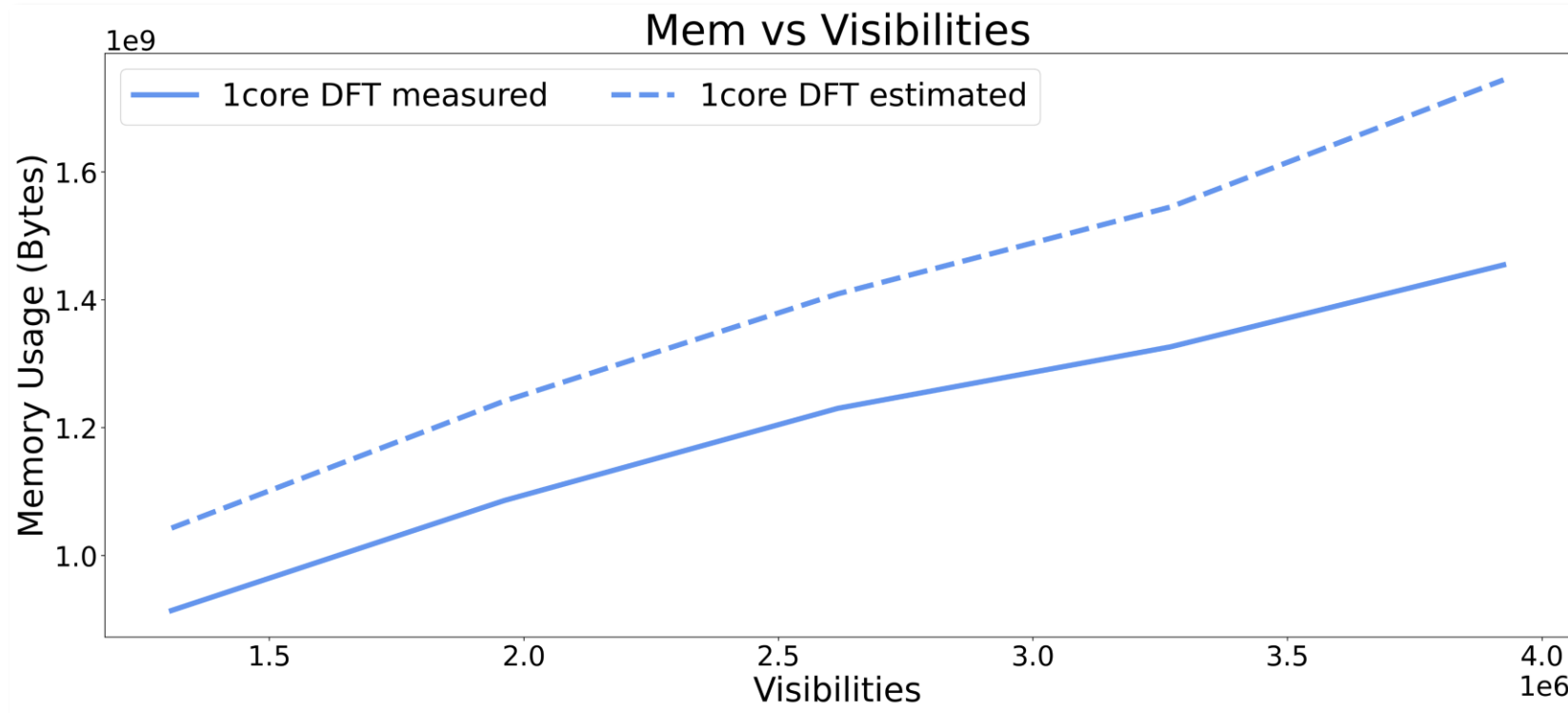
Use PREESM's reported allocated inter-node memory as estimation

```
Starting allocation with BestFitAllocator(LARGEST_FIRST)  
BestFitAllocator(LARGEST_FIRST) allocates 3.2940004020929337 GBytes in 171 ms.  
Workflow Step: Code Generation (id: org.iefr.preesm.codegen.xtend.task.Codege
```

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
```

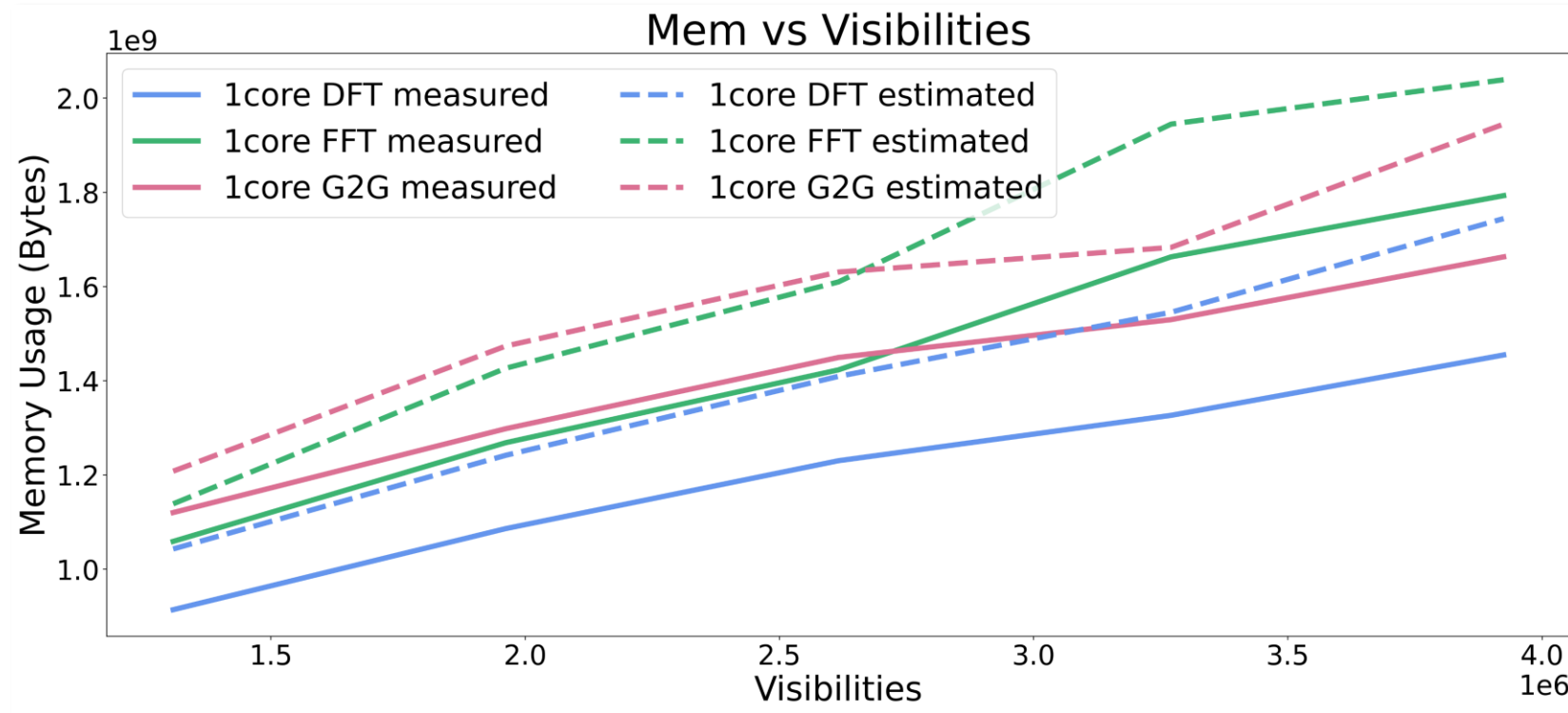
Resource Estimation – Memory: Results



Predicts the general trend.

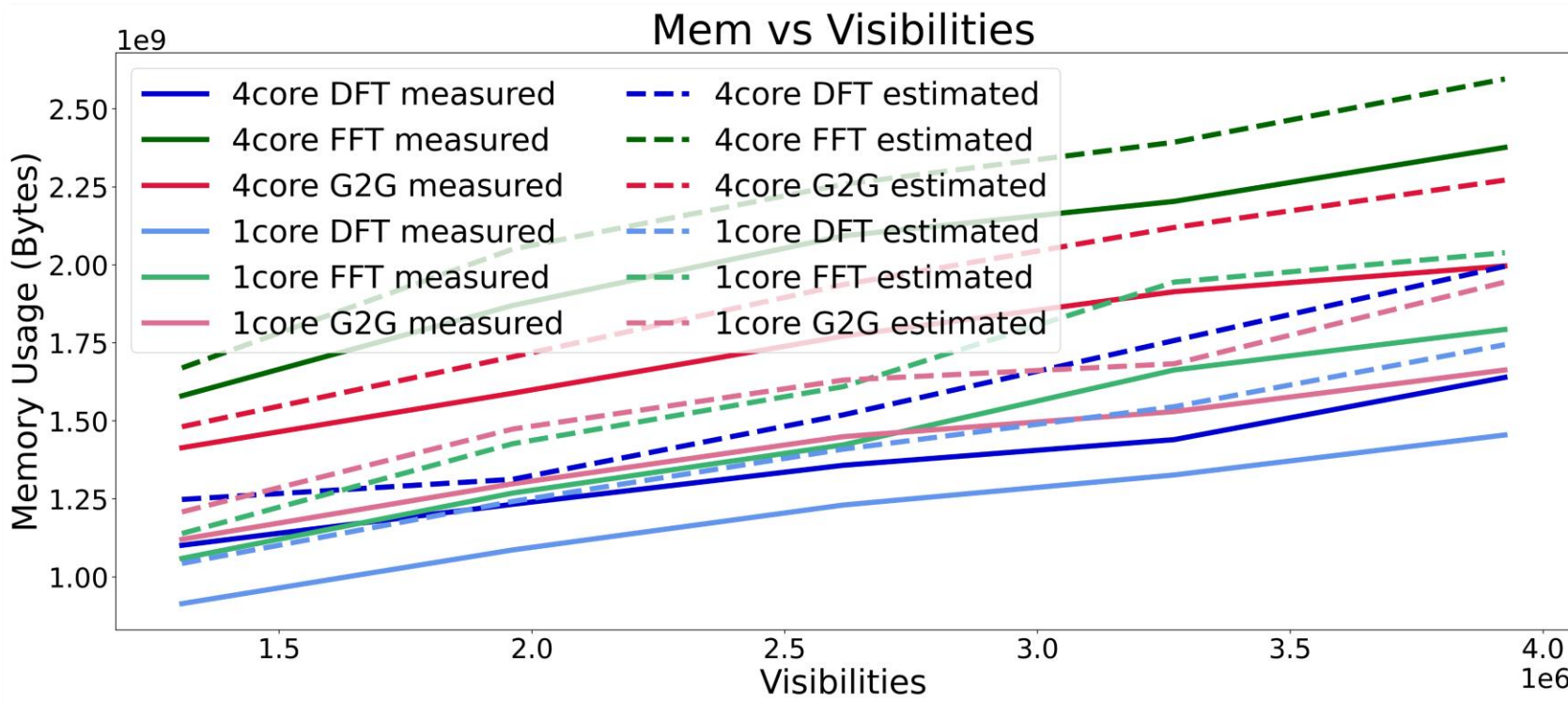
Unexpected that measured < estimated (should be the inverse). Need to investigate

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

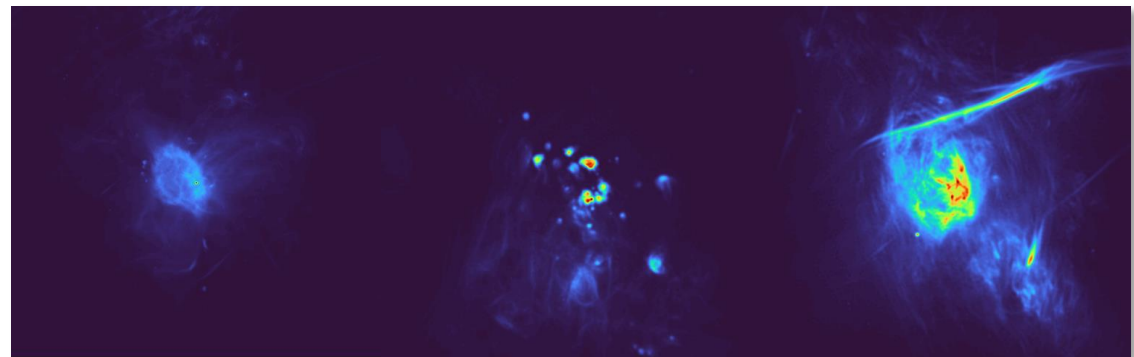
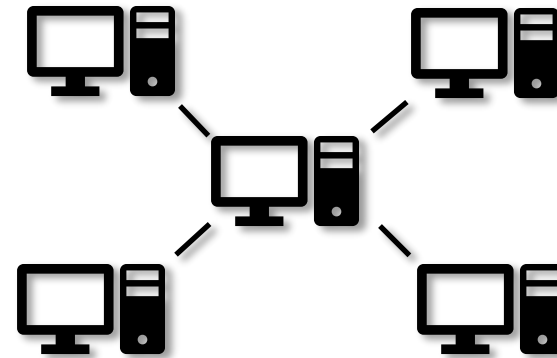
Main drawback is that only static analysis is supported currently

- 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?