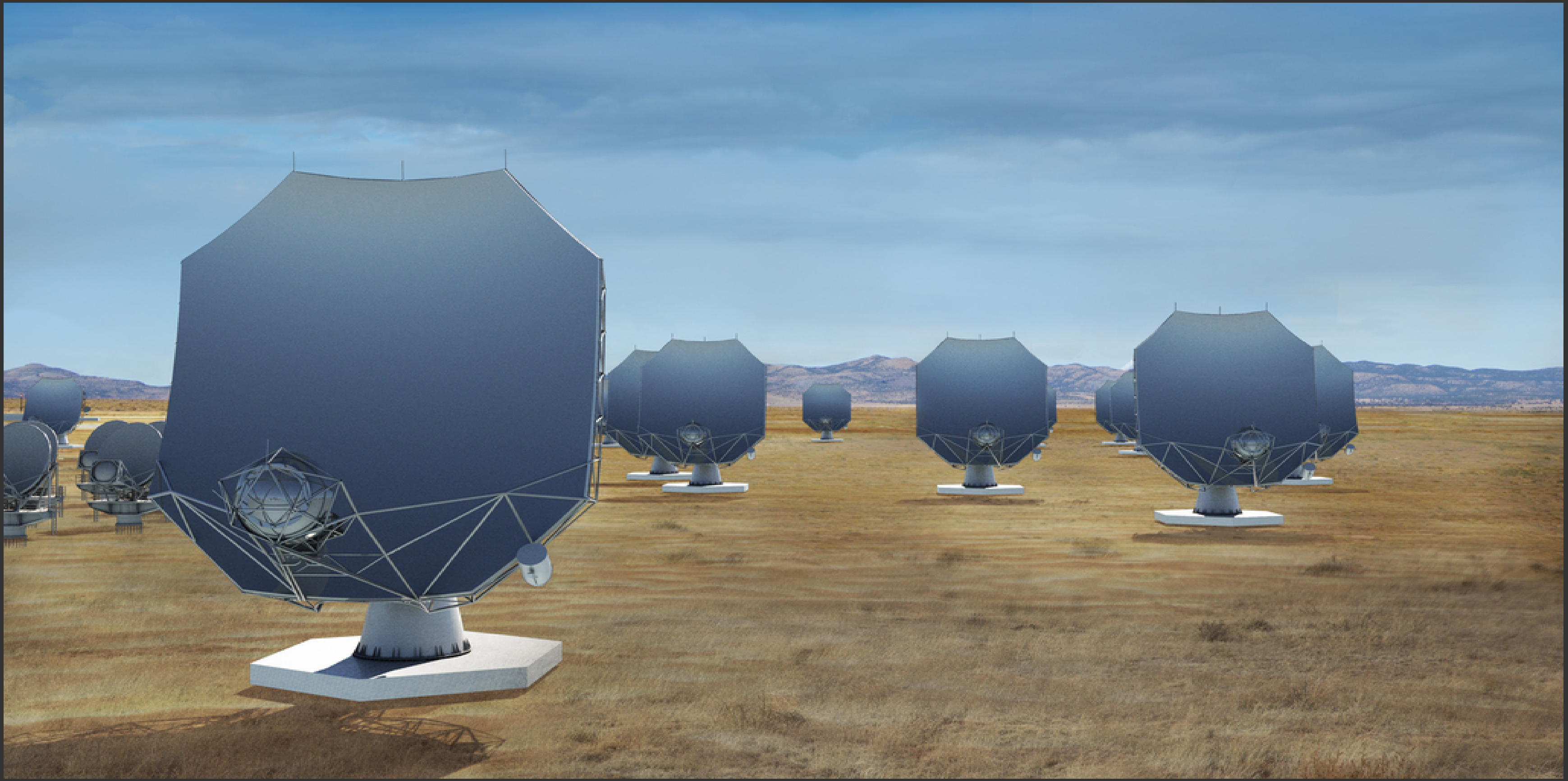
Data Processing Landscape

PetaFLOPs, TeraBytes & Algorithms

Wed. Lunch, Socorro, NM, May 8th 2024



S. Bhatnagar

Algorithms R&D Group,

National Radio Astronomy Observatory, Socorro, NM, USA

NRAO

The Algorithms R&D Group (ARDG)

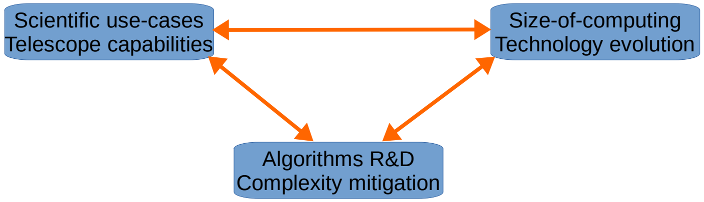
- Current membership
    - Sanjay Bhatnagar (Lead) (50%)
    - Preshanth Jagannathan (Assist. Sci.) (50% ARDG, 25% CASA, 25% ngVLA CIPT)
    - Genie Hsieh (Software Eng.) (100%)
    - Felipe Madsen (100%)

- Total effort
    - ~2.5 FTE from 4 full-time staff
    - Mentoring 1 Jy PDF (Hendrik Mueller)
    - Collaborations
        - NRAO SIS Group
        - External groups/industry:

          Kokkos(SNL/DoE), CHTC/PATh, DSA2K/CalTech, NVIDIA

Summary

- Inherent complexity

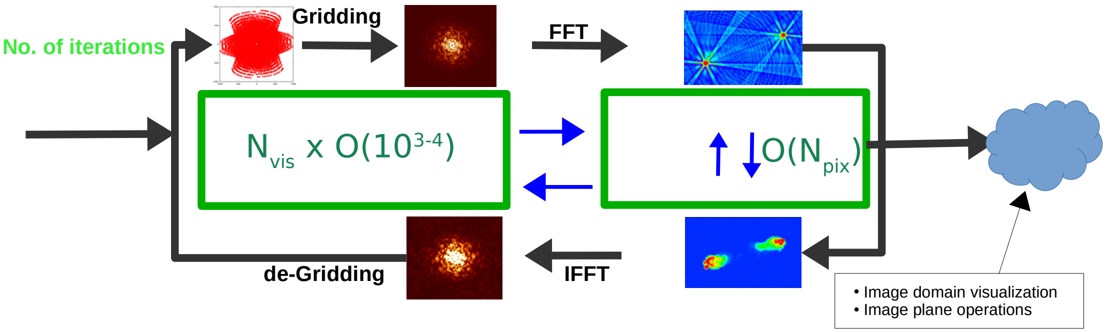


- Size of computing and the projected technology landscape
  - Scalable algorithm and software architecture
- Algorithms R&D
  - A reliable, stable software system
  - Algorithms for faster convergence, impact overall cost of computing
- Collaborations: HTC, HPC, industry groups, learn from current literature,...
  - Scaling on larger, externally managed heterogeneous clusters
  - Impact on R&D, s/w design, management,...

System level description

- Typical data processing steps

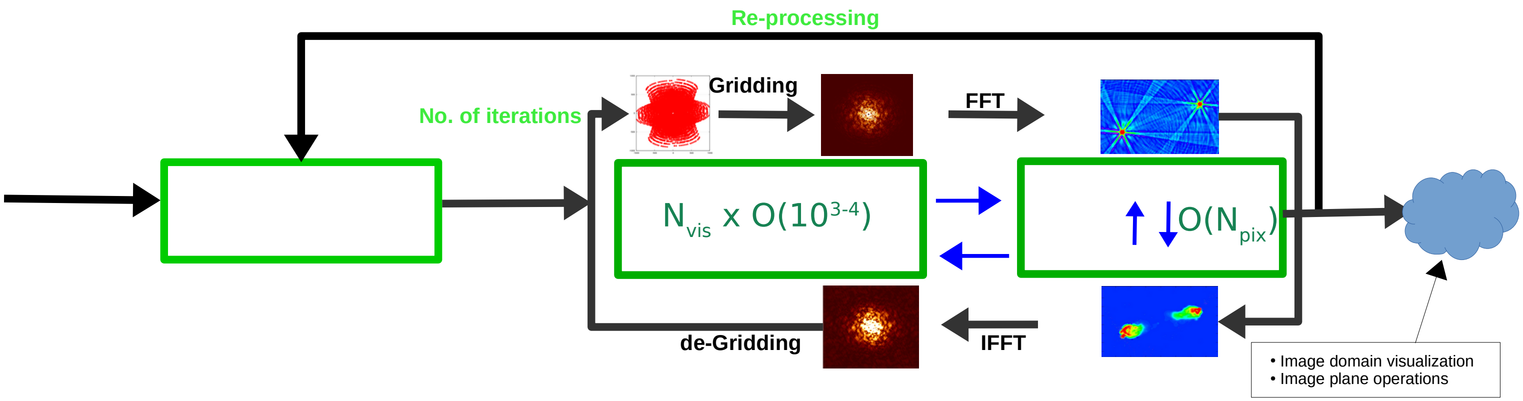
Imaging: N_{vis} x $O(10^{3-4})$ FLOPs (Complex, SP + DP)
Image-plane deconvolution of the PSF : $O(N_{pix})$ FLOPs (Real-valued, SP)



System level description

- Typical data processing steps

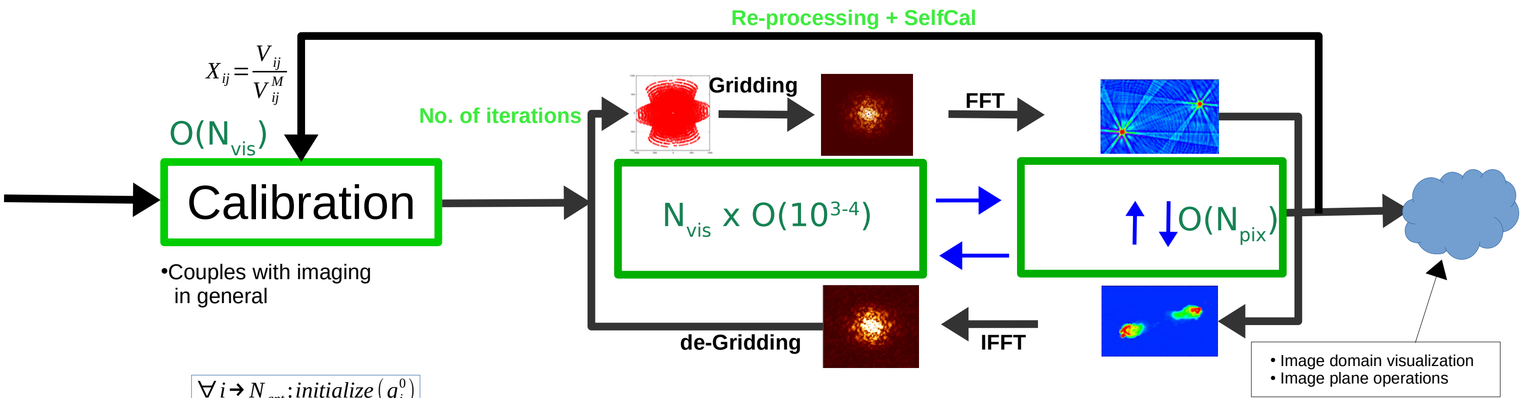
Imaging: N_{vis} x $O(10^{3-4})$ FLOPs (Complex, SP + DP)
Image-plane deconvolution of the PSF : $O(N_{pix})$ FLOPs (Real-valued, SP)



System level description

- Typical data processing steps

Imaging:	N_{vis} x $O(10^{3-4})$	FLOPs (Complex, SP + DP)
Image-plane deconvolution of the PSF :	$O(N_{pix})$	FLOPs (Real-valued, SP)
Calibration:	$O(N_{vis})$	FLOPs (Complex, SP)



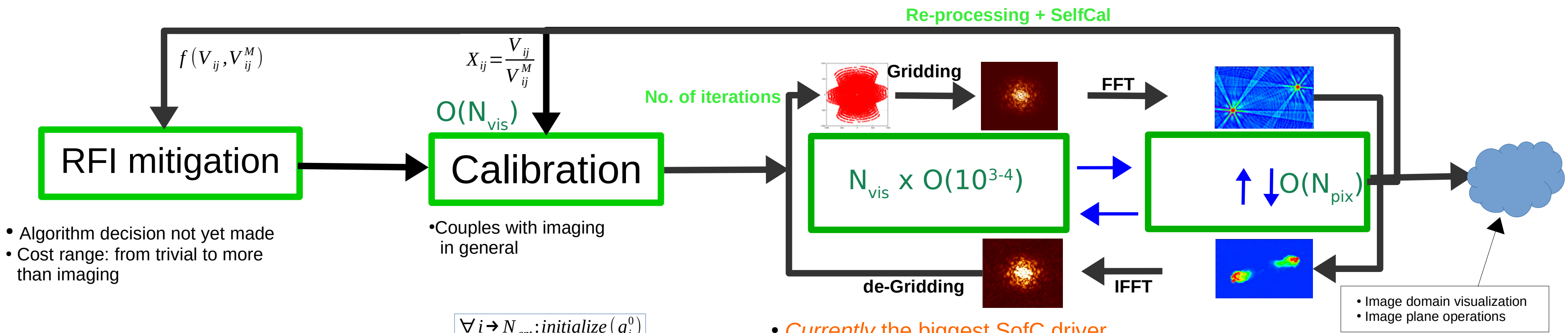
$$\forall i \rightarrow N_{ant} : initialize\left(g_i^0\right)$$
$$\forall i \rightarrow N_{ant} :$$
$$d = n = 0.0$$
$$\forall j \rightarrow N_{ant} :$$
$$d = d + g_j^{n-1} X_{ij} W_{ij}$$
$$n = n + \left|g_j^{n-1}\right|^2 W_{ij}$$
$$g_i^n = g_i^{n-1} + \alpha \frac{d}{n}$$

Wed. Lunch, Socorro, May 8th 2024

System level description

- Typical data processing steps

Imaging:	N_{vis} x $O(10^{3\text{-}4})$ FLOPs (Complex, SP + DP)
Image-plane deconvolution of the PSF :	$O(N_{pix})$ FLOPs (Real-valued, SP)
Calibration:	$O(N_{vis})$ FLOPs (Complex, SP)
Flagging:	Trivial → dominant!

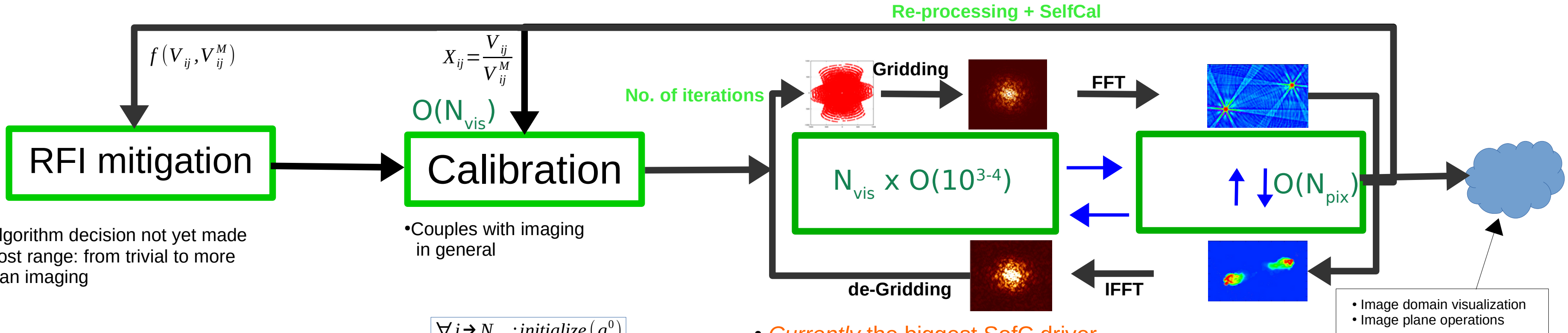


$$
\begin{aligned}
&\forall i \rightarrow N_{ant} : initialize\,(g_i^0) \\
&\forall i \rightarrow N_{ant} : \\
&\quad d = n = 0.0 \\
&\quad \forall j \rightarrow N_{ant} : \\
&\qquad d = d + g_j^{n-1} X_{ij} W_{ij} \\
&\qquad n = n + |g_j^{n-1}|^2 W_{ij} \\
&\quad g_i^n = g_i^{n-1} + \alpha \frac{d}{n}
\end{aligned}
$$

System level description

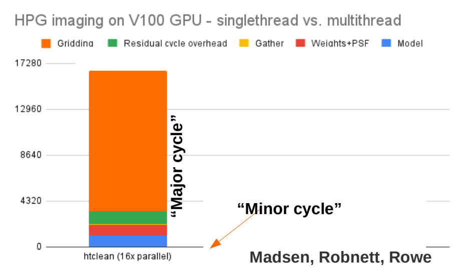
- Typical data processing steps

Imaging:	N_{vis} x O($10^{3\text{-}4}$) FLOPs (Complex, SP + DP)
Image-plane deconvolution of the PSF :	O(N_{pix}) FLOPs (Real-valued, SP)
Calibration:	O(N_{vis}) FLOPs (Complex, SP)
Flagging:	Trivial → dominant!



$$\forall i \rightarrow N_{ant} : initialize(g^0_i)$$
$$\forall i \rightarrow N_{ant} :$$
$$d = n = 0.0$$
$$\forall j \rightarrow N_{ant} :$$
$$d = d + g^{n-1}_j X_{ij} W_{ij}$$
$$n = n + |g^{n-1}_j|^2 W_{ij}$$
$$g^n_i = g^{n-1}_i + \alpha \frac{d}{n}$$

- *Currently* the biggest SofC driver



Madsen, Robnett, Rowe

Size of computing

- Dominated by the imaging operation

https://library.nrao.edu/public/memos/ngvla/NGVLAC_04.pdf

$$CL_{\mathrm{WP}} = [N_{Overhead} FLOPS] \sum_{w=0}^{W_{max}-1} N_{vis}(w) [S(w=0)(\alpha w^2 + 1)]^2$$

$$CL_{AP} = [N_{Overhead} FLOPS] \sum_{i=0}^{N_{spw}-1} N_{vis}(\nu_i) [S(\nu_o) \frac{\nu_i}{\nu_o}]^2$$

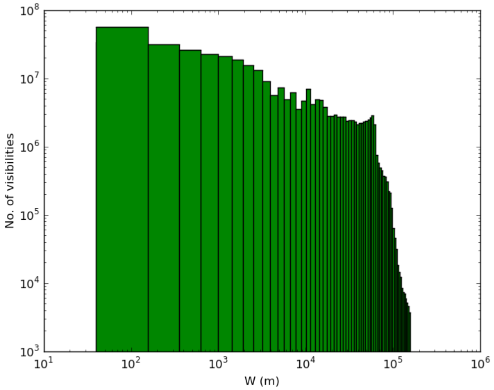
Size of computing

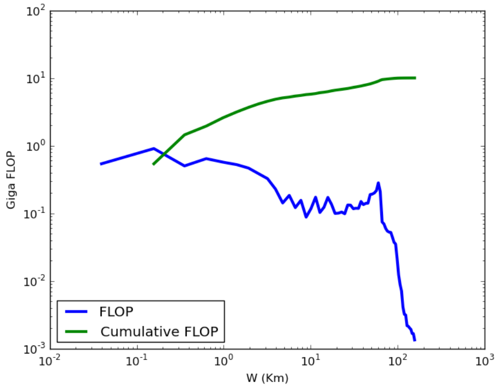
- Dominated by the imaging operation

https://library.nrao.edu/public/memos/ngvla/NGVLAC_04.pdf

$$CL_{\mathrm{WP}} = [N_{Overhead}\, FLOPS] \sum_{w=0}^{W_{max}-1} N_{vis}(w) [S(w=0)(\alpha w^2+1)]^2$$

$$CL_{AP} = [N_{Overhead}\, FLOPS] \sum_{i=0}^{N_{spw}-1} N_{vis}(\nu_i) [S(\nu_o) \frac{\nu_i}{\nu_o}]^2$$





Size of computing

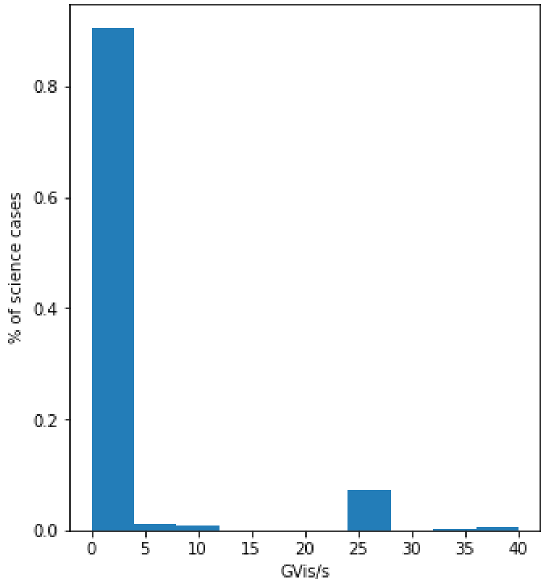
- Dominated by the imaging operation

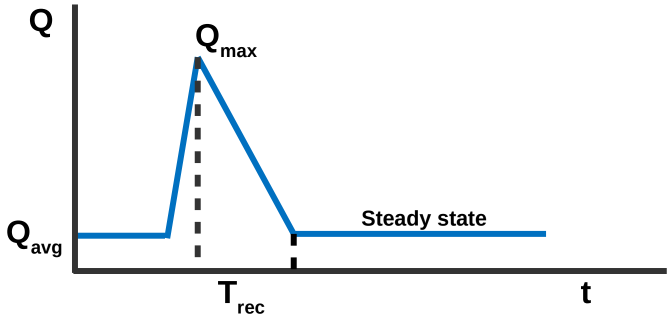
https://library.nrao.edu/public/memos/ngvla/NGVLAC_04.pdf

$$CL_{\mathrm{WP}} = [N_{Overhead} \, FLOPS] \sum_{w=0}^{W_{max}-1} N_{vis}(w) [S(w=0)(\alpha w^2 + 1)]^2$$

$$CL_{AP} = [N_{Overhead} \, FLOPS] \sum_{i=0}^{N_{spw}-1} N_{vis}(\nu_i) [S(\nu_o) \frac{\nu_i}{\nu_o}]^2$$

$$SP_K = w_K \frac{\kappa_K \, CL_K}{\epsilon_c \, \epsilon_p} \; FLOPS/sec$$





Size of computing

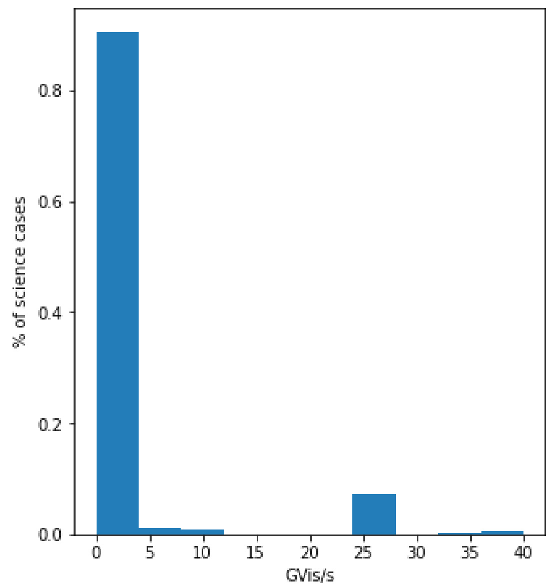
- Dominated by the imaging operation

https://library.nrao.edu/public/memos/ngvla/NGVLAC_04.pdf

$$CL_{WP}=[N_{Overhead}\,FLOPS]\sum_{w=0}^{W_{max}-1} N_{vis}(w)[S(w=0)(\alpha w^2+1)]^2$$

$$CL_{AP}=[N_{Overhead}\,FLOPS]\sum_{i=0}^{N_{spw}-1} N_{vis}(\nu_i)[S(\nu_o)\frac{\nu_i}{\nu_o}]^2$$

$$SP_K = w_K \frac{\kappa_K\, CL_K}{\epsilon_c\,\epsilon_p}\ FLOPS/sec$$



- ngVLA: 50 PFLOP/sec (**T_{rec}** ~1 day) ← ~~O(million) CPU cores/~~few x O(1000) GPUs
- WSU: O(100) TFLOP/sec ← Can be done today!

Size of computing

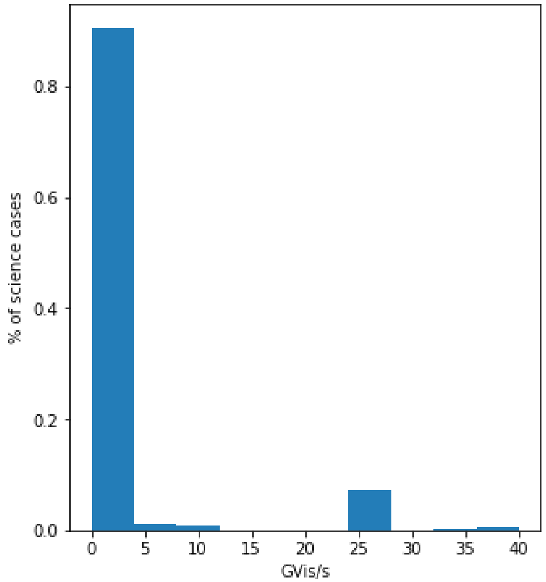
- Dominated by the imaging operation

https://library.nrao.edu/public/memos/ngvla/NGVLAC_04.pdf

$$CL_{WP} = [N_{Overhead}\, FLOPS] \sum_{w=0}^{W_{max}-1} N_{vis}(w)[S(w=0)(\alpha w^2+1)]^2$$

$$CL_{AP} = [N_{Overhead}\, FLOPS] \sum_{i=0}^{N_{spw}-1} N_{vis}(\nu_i)[S(\nu_o)\frac{\nu_i}{\nu_o}]^2$$

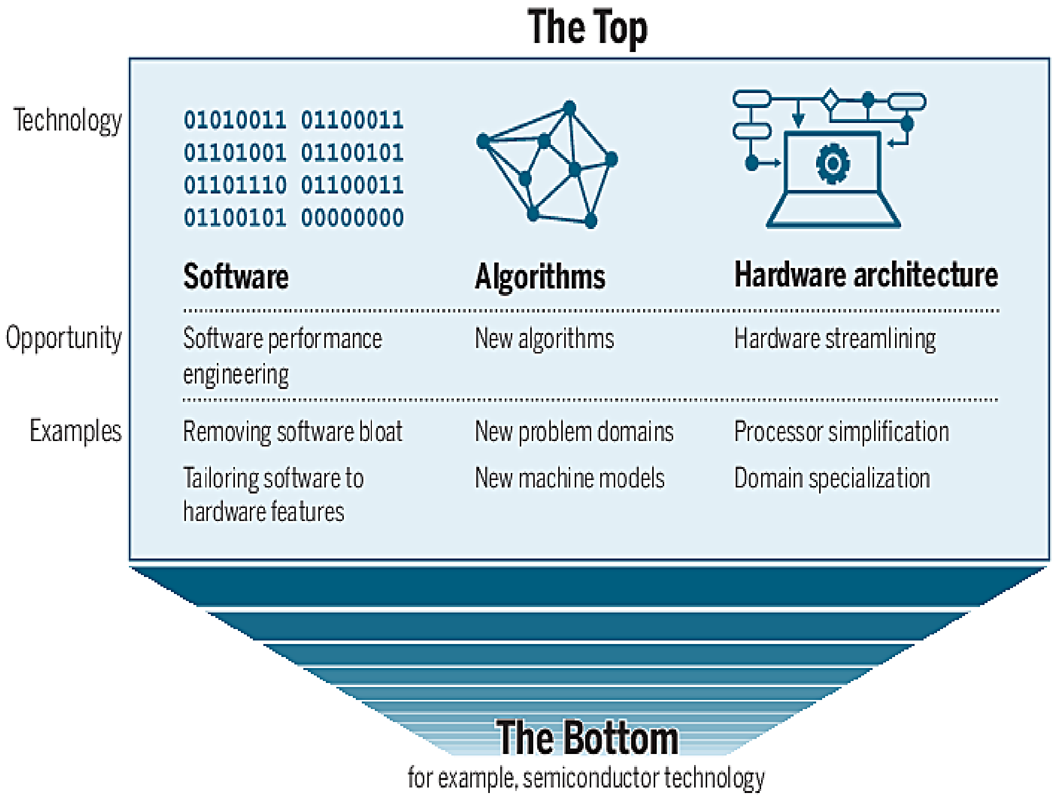
$$SP_K = w_K \frac{\kappa_K\, CL_K}{\epsilon_c\, \epsilon_p}\ FLOPS/sec$$



!!

- ngVLA: 50 PFLOP/sec **(T_{rec}** ~1 day) ← ~~O(million) CPU cores/~~few x O(1000) GPUs
- WSU: O(100) TFLOP/sec ← Can be done today!

Computing stack: Room at the Top

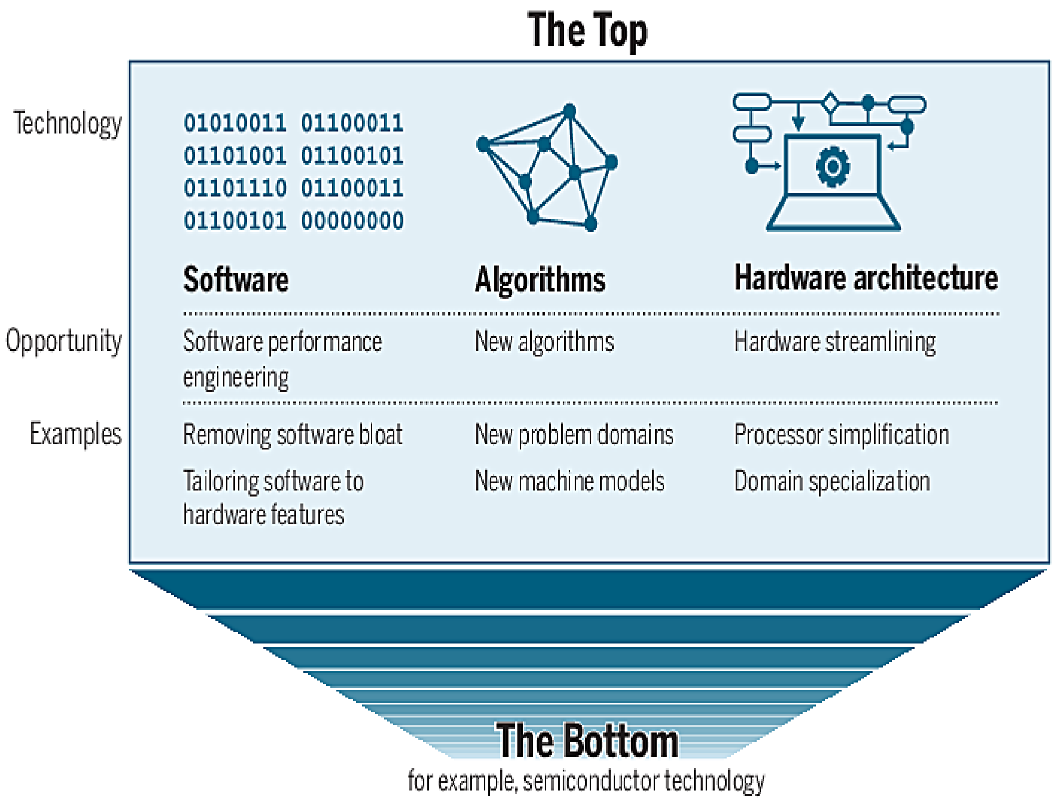


"There is room at the Bottom"
- Feynman (1959)

Performance gains after Moore's law ends. In the post-Moore era, improvements in computing power will increasingly come from technologies at the "Top" of the computing stack, not from those at the "Bottom", reversing the historical trend.

Leiserson et al. Science (2020)

Computing stack: Room at the Top



Performance gains after Moore's law ends. In the post-Moore era, improvements in computing power will increasingly come from technologies at the "Top" of the computing stack, not from those at the "Bottom", reversing the historical trend.

Leiserson et al. Science (2020)

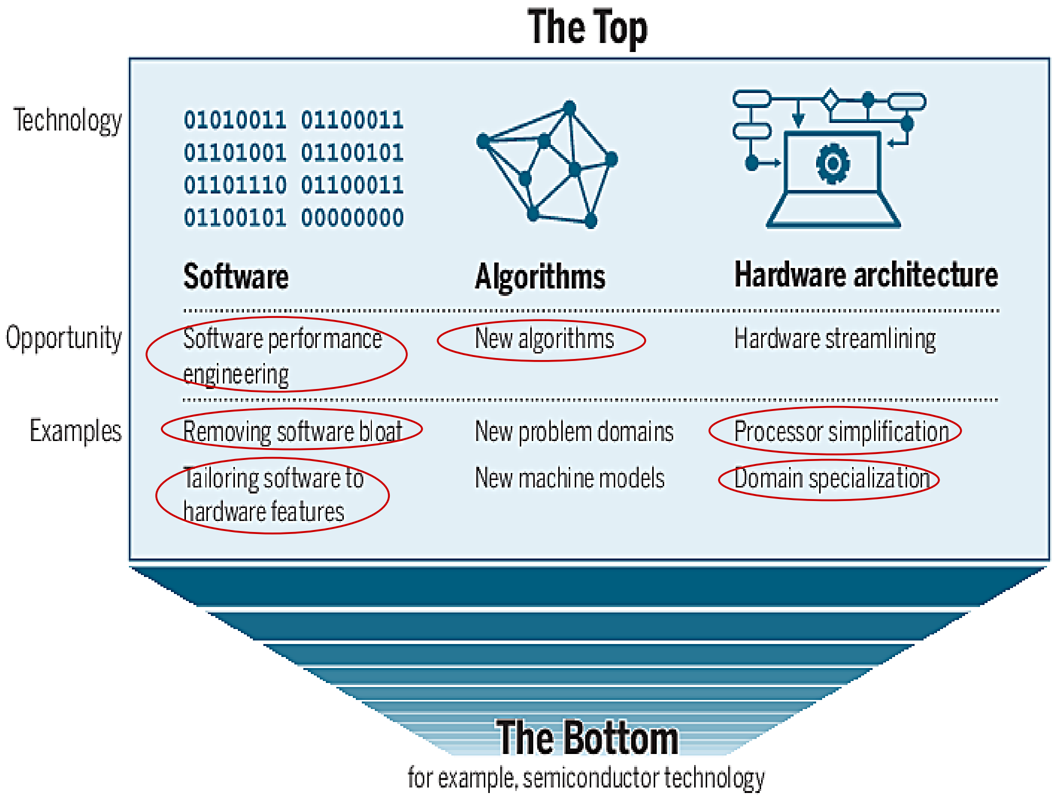
"There's plenty of room at the Top"
- Leiserson et al. (2020)

Current capacity utilization:
In single-digit percentage

Need algorithms with high
Arithmetic Intensity (FLOPs per byte)

"There is room at the Bottom"
- Feynman (1959)

Computing stack: Room at the Top



Performance gains after Moore's law ends. In the post-Moore era, improvements in computing power will increasingly come from technologies at the "Top" of the computing stack, not from those at the "Bottom", reversing the historical trend.

Leiserson et al. Science (2020)

"There's plenty of room at the Top"

- Leiserson et al. (2020)

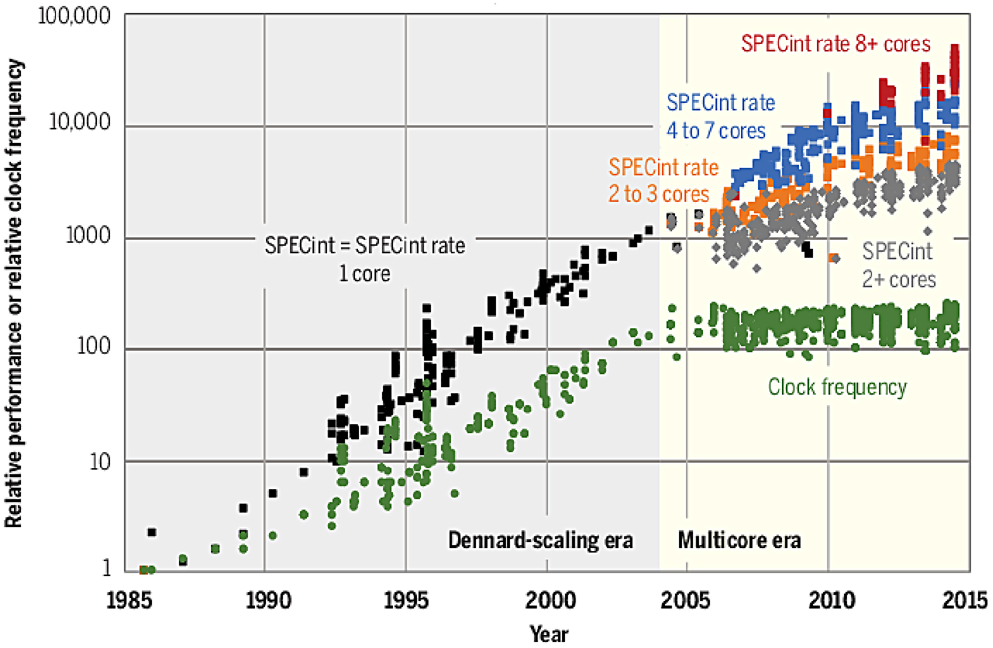
Current capacity utilization:
In single-digit percentage

Need algorithms with high
Arithmetic Intensity (FLOPs per byte)

"There is room at the Bottom"

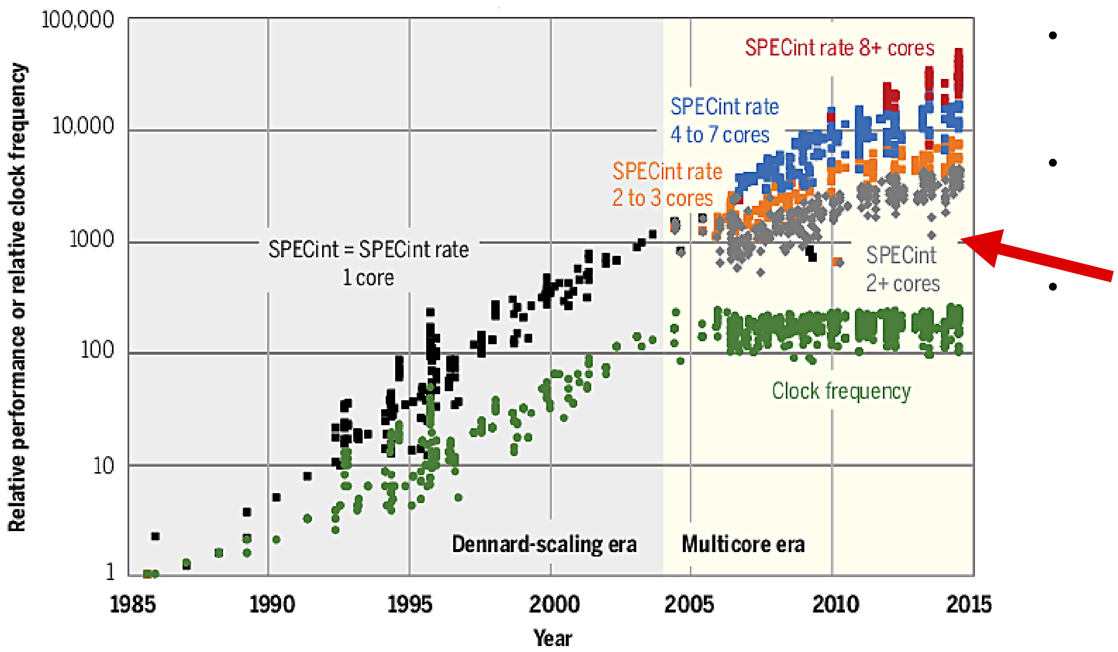
- Feynman (1959)

Computing stack: Multi-core era



- Moore's Law era
  - Runtime reduced by 2x if one just waited
  - Improvements were more predictable

Computing stack: Multi-core era

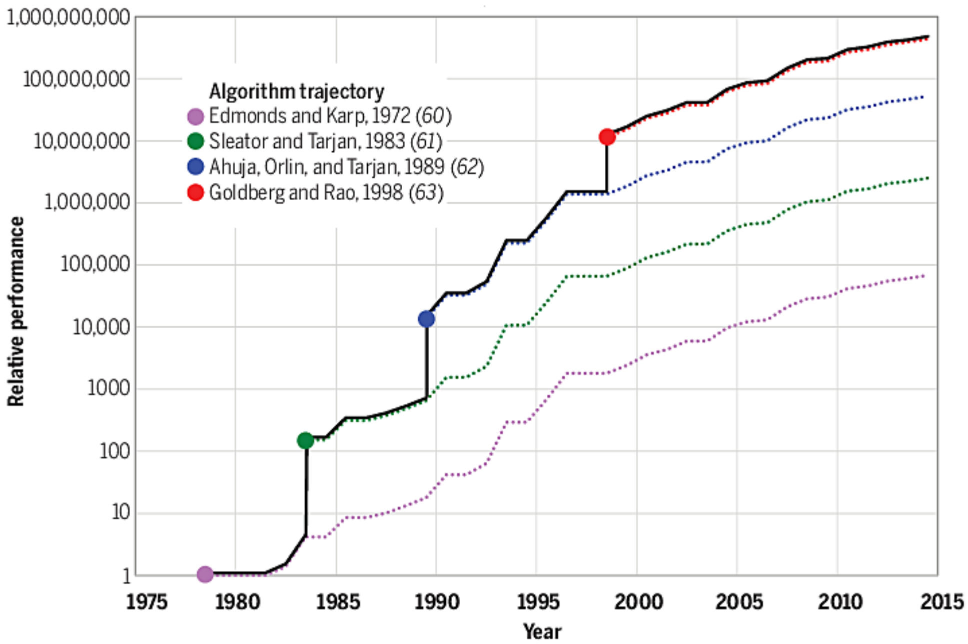


- ~~Moore's Law era~~
  - ~~Runtime reduced by 2x if one just waited~~
  - ~~Improvements were more predictable~~
- Number of GP cores now is also limited by the end of Moore's-law era.
- The Top: Post Moore's-Law era:

  Massively parallel h/w of simpler cores (not GP)

  - Improvements from performance engineering, new algorithms, better silicon utilization
  - Algorithms that effectively parallelize on multiple scales of the problem
  - Specialized software

Computing stack: Algorithms



- Historically AR&D has delivered runtime gains comparable to the Moore's Law
- Moore's Law has historically caught up...but that has now ended!

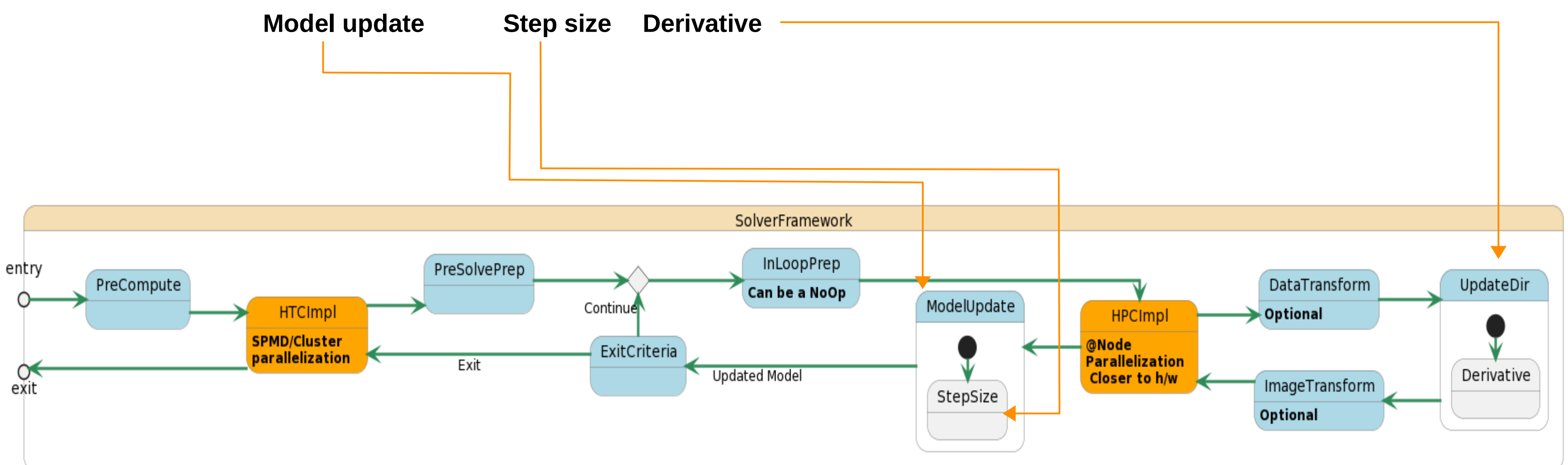
- RA algorithms have a higher FLOP per byte ratio
- RA problem: Combination of HPC (PetaFLOPs) + Big Data (TeraBytes) + 24x7 operation (High Throughput)

Algorithm Architecture

- Stable, Scalable Architecture
  - Must scale with evolving computing needs (std VLA vs VLASS), algorithms, computing h/w & s/w (heterogeneous cluster)
  - Cast our algorithms in standard terminology: Derivative, Hessian, Update,...
  - Decompose into functionally separable components which can scale individually and together

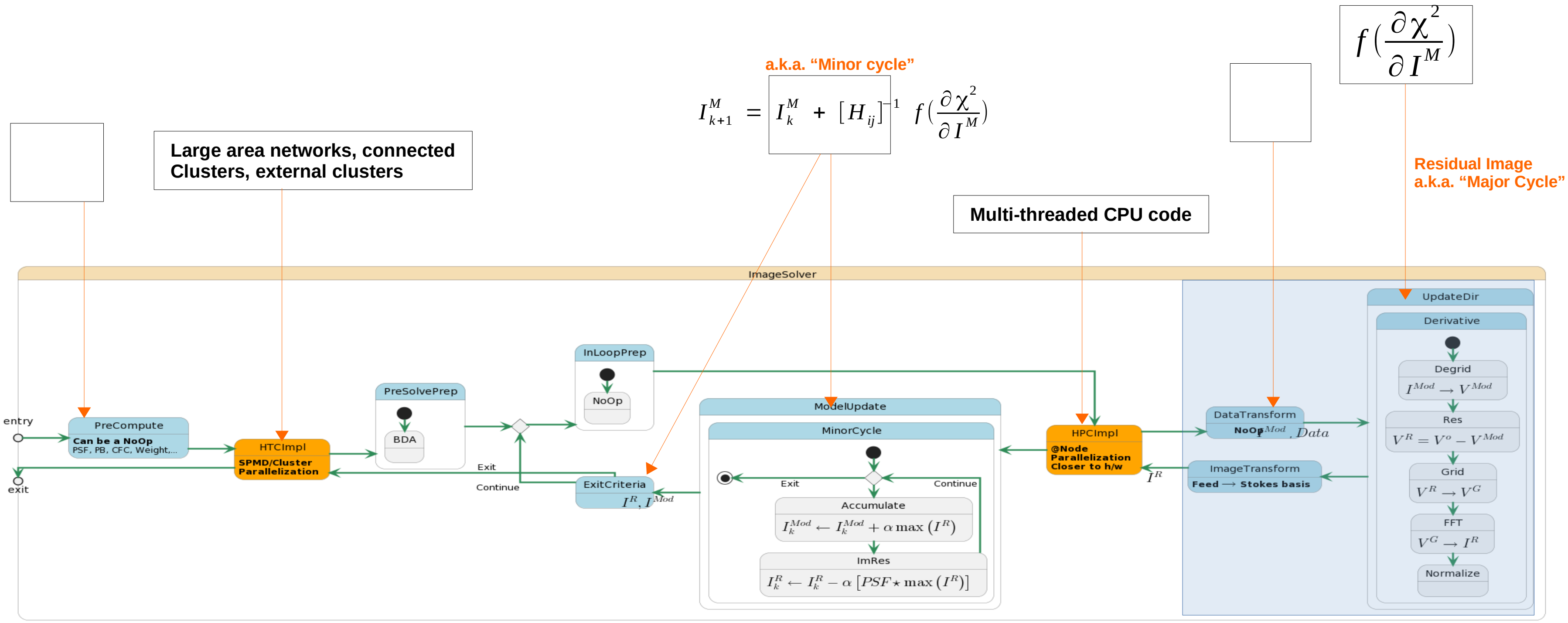
$$V^{obs} = \mathbf{G}^M S \mathbf{F} B^M I^M + noise \qquad \chi^2 = \sum_i \left|Data_i - Model_i(P)\right|^2$$

$$P_i^{k+1} = P_i^k + [H_{ij}]^{-1} f\left(\frac{\partial \chi^2}{\partial P_i^k}\right) \quad ; \qquad [H_{ij}] = \frac{\partial^2 \chi^2}{\partial P_i^k \partial P_j^k}$$



Algorithm Architecture: Imaging

- Mathematical framework is the same for calibration and imaging
- Specialization of the components delivers various calibration and imaging algorithms



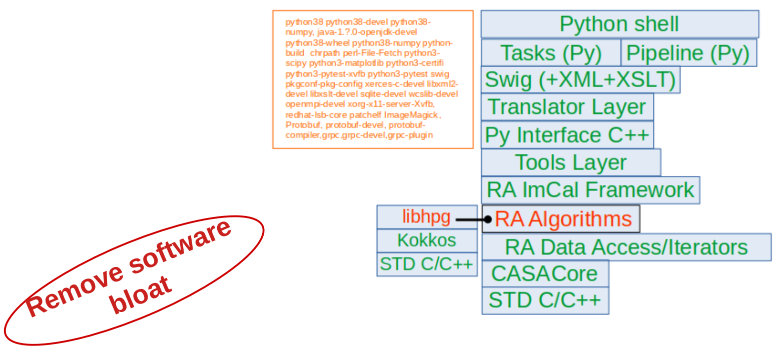
The LibRA Project: By the users, for the users

- Goals: Re-use code, re-usable library, relocatable software, ease of use
  - Derived from CASAScientific. Now an independent code base + build system
  - Enable collaborations with RA groups and end-users + with other domains: HPC, HTC, Medical imaging,…

    https://github.com/ARDG-NRAO/LibRA

- Directly use the scientific layer via standalone applications
  - Deployable on external heterogeneous cluster of CPUs + GPUs

- Automate chores: Modernized build system, containerized deployment, Py binding,…



Architectural components as standalone relocatable apps

```
>roadrunner
vis               = VLASS2.1.sb38453816.eb38509426.59047.17567765046_split.ms
imagename         = refim_oneshiftpoint.res
modelimagename    =
datacolumn        = data
sowimageext       = sumwt
complexgrid       =
imsize            = 16384
cell              = 0.6
stokes            = I
reffreq           = 3.0GHz
phasecenter       = 22:10:0.000 -00.30.0.0000 J2000
weighting         = natural
wprojplanes       = 1
gridder           = awphpg
cfcache           = w1.cf
mode              = residual
wbawp             = 1
field             =
spw               = 2~17
uvrange           =
pbcor             = 1
conjbeams         = 0
pblimit           = 0.001
usepointing       = 0
roadrunner>
```

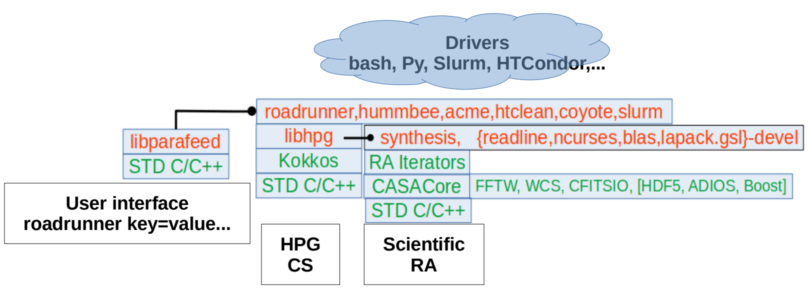
The LibRA Project: By the users, for the users

- Goals: Re-use code, re-usable library, relocatable software, ease of use
  - Derived from CASAScientific. Now an independent code base + build system
  - Enable collaborations with RA groups and end-users + with other domains: HPC, HTC, Medical imaging,...

    https://github.com/ARDG-NRAO/LibRA

- Directly use the scientific layer via standalone applications
  - Deployable on external heterogeneous cluster of CPUs + GPUs

- Automate chores: Modernized build system, containerized deployment, Py binding,...

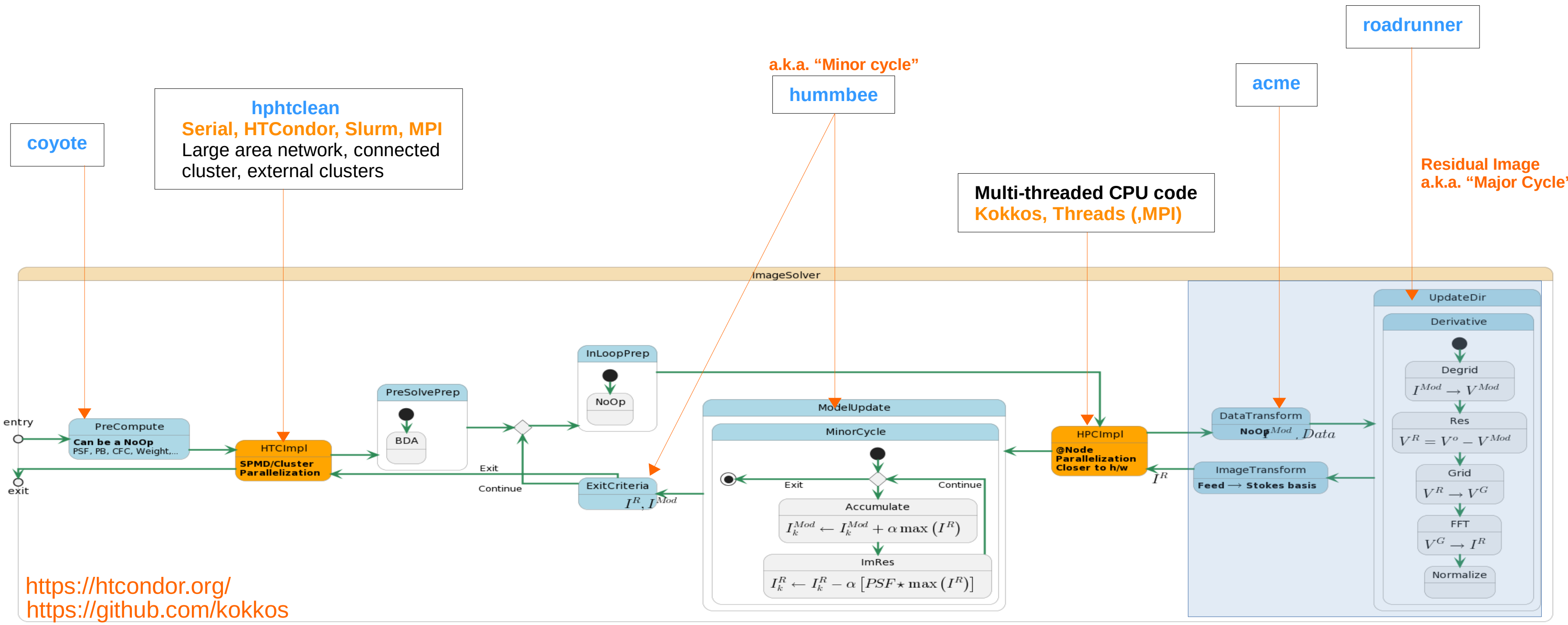


Architectural components as standalone relocatable apps

```
>roadrunner
vis                  = VLASS2.1.sb38453816.eb38509426.59047.17567765046_split.ms
imagename            = refim_oneshiftpoint.res
modelimagename       =
datacolumn           = data
sowimageext          = sumwt
complexgrid          =
imsize               = 16384
cell                 = 0.6
stokes               = I
reffreq              = 3.0GHz
phasecenter          = 22:10:0.000 -00.30.0.0000 J2000
weighting            = natural
wprojplanes          = 1
gridder              = awphpg
cfcache              = w1.cf
mode                 = residual
wbawp                = 1
field                =
spw                  = 2~17
uvrange              =
pbcor                = 1
conjbeams            = 0
pblimit              = 0.001
usepointing          = 0
roadrunner>
```

Algorithm Architecture: Deployment

- Mathematical framework is the same for calibration and imaging
- Specialization of the components delivers various calibration and imaging algorithms



High Performance Gridder (HPG)

- A gridder/de-gridder that runs on a GPUs, multi-threaded on CPUs
- Built on the Kokkos framework: Choice based on projected technology evolution
  - Implemented as a reusable independent library (ngVLA Comp. Memo #4, #5, #7)

Tailoring software to Hardware features

High Performance Gridder (HPG)

- A gridder/de-gridder that runs on a GPUs, multi-threaded on CPUs
- Built on the Kokkos framework: Choice based on projected technology evolution
  - Implemented as a reusable independent library (ngVLA Comp. Memo #4, #5, #7)
- Algorithm parameterized by scientific use-cases and their evolution.

$$V_{ij}^{G}=\left[M_{ij}\,e^{\iota(\vec{\phi}_{ij}+\vec{\theta}^{M})\cdot\Delta\vec{B}}\right]*V_{ij}^{o} \qquad I=FFT\left(V^{G}\right)$$

$$M_{ij}=CF=ATerm*WTerm*PSTerm$$

  - Configurable: (Single pointing, Pointed mosaic, OTF mosaic) + Antenna pointing corrections

Tailoring software to Hardware features

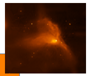
Operation	ATerm	PSTerm	WTerm wprojplanes	CF
AW-Projection	True	True False	>1	PS*A*W A*W
A-Projection	True	True False	1	PS*A A
W-Projection	False	True	>1	PS*W
Standard	False	True	1	PS

EVLA Memo 84 (2004)
AJ, V. 154,#5 (2017)
ApJ,Vol.770, No. 2, 91 (2013)
A&A 487, 419-429 (2008)

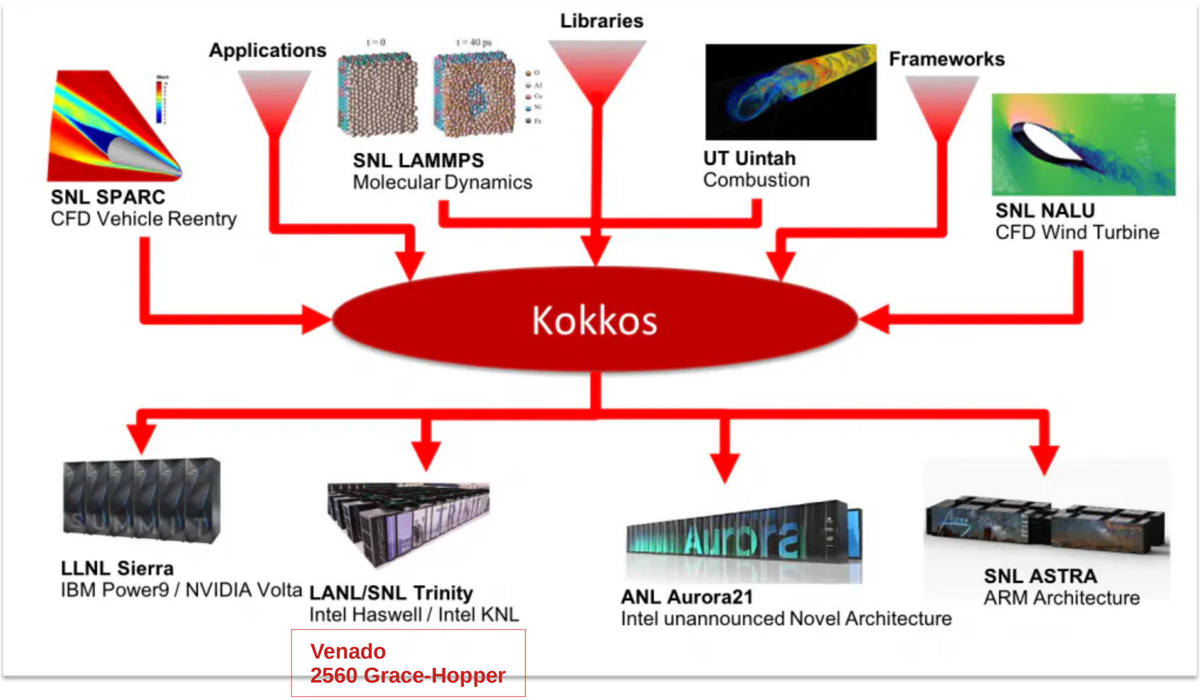
Kokkos: Performance portable eco-system (DoE)

roadrunner

LibRA



LibRA framework

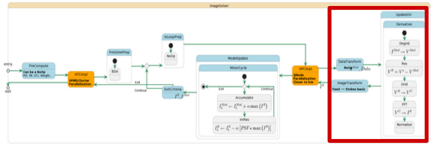


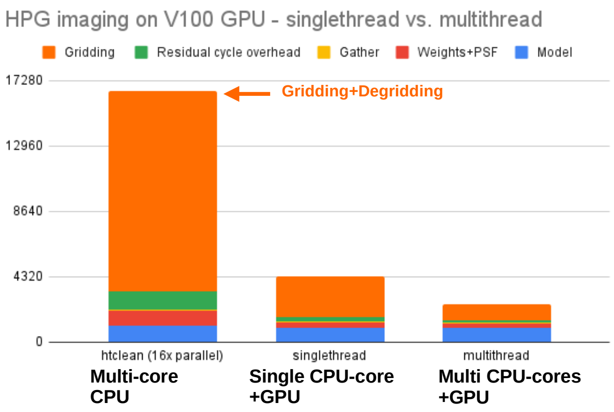
Software performance engineering

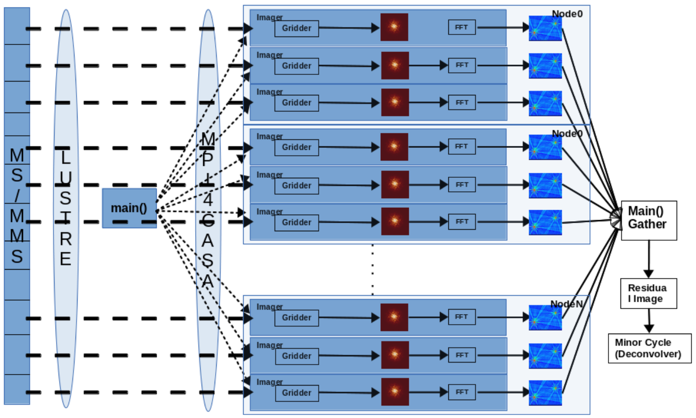
- C++ Performance Portability Ecosystem is a production level solution for writing modern C++ applications in a hardware agnostic way.
- Part of the US Department of Energies Exascale Project – the leading effort in the US to prepare the HPC community for the next generation of super computing platforms.

HPG characterization

- Measured speed-up: 100 – 200x compared to a single CPU core

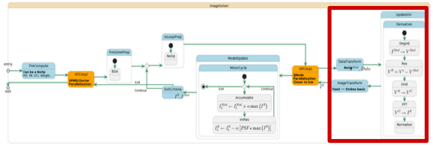


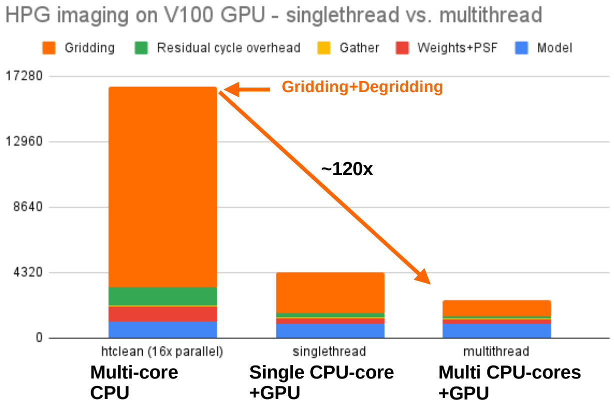


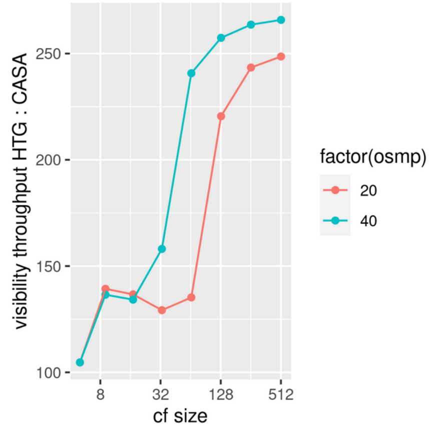


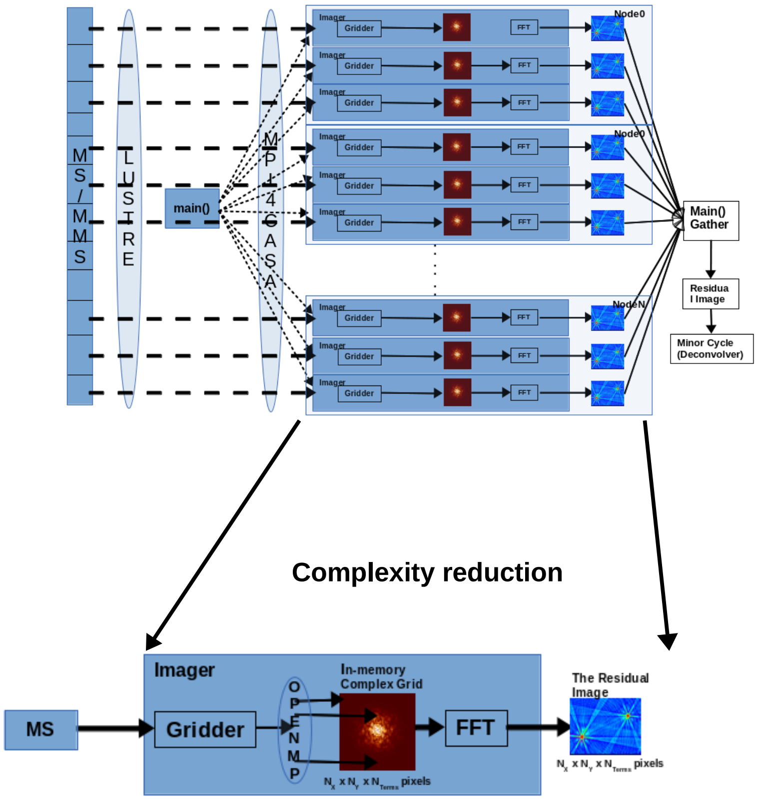
HPG characterization

- Measured speed-up: 100 – 200x compared to a single CPU core





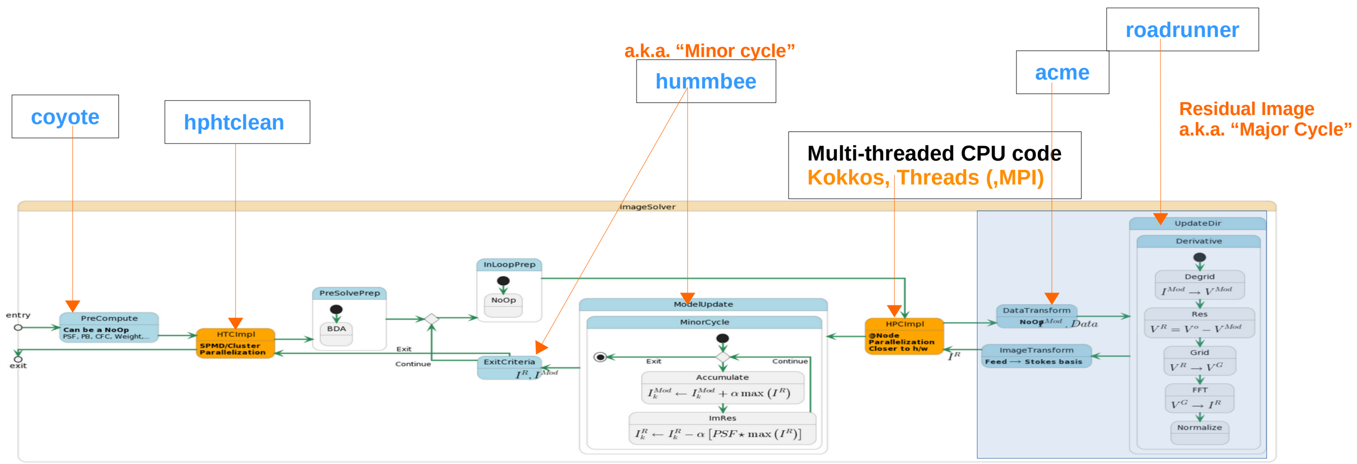




$$CL_{WP}=[N_{Overhead}FLOPS]\sum_{w=0}^{W_{max}-1}N_{vis}(w)[S(w=0)(\alpha w^2+1)]^2$$

Throughput measurements

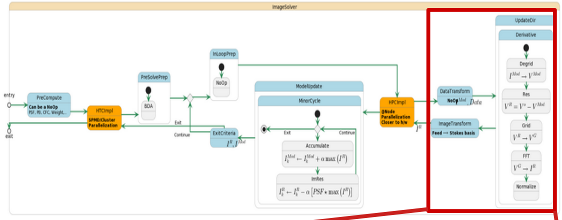
- Deployed on a cluster of GPUs (100) on the PATh facility in collaboration with https://science.nrao.edu/enews/17.3/index.shtml#deepimaging
  - The Center for High Throughput Computing (CHTC, UW-M)
  - National Research Platform (NRP) via the OSPool, Nebraska node
  - The San Diego Super Computer Center (SDSC)
  - + Multiple university computer centers across the US

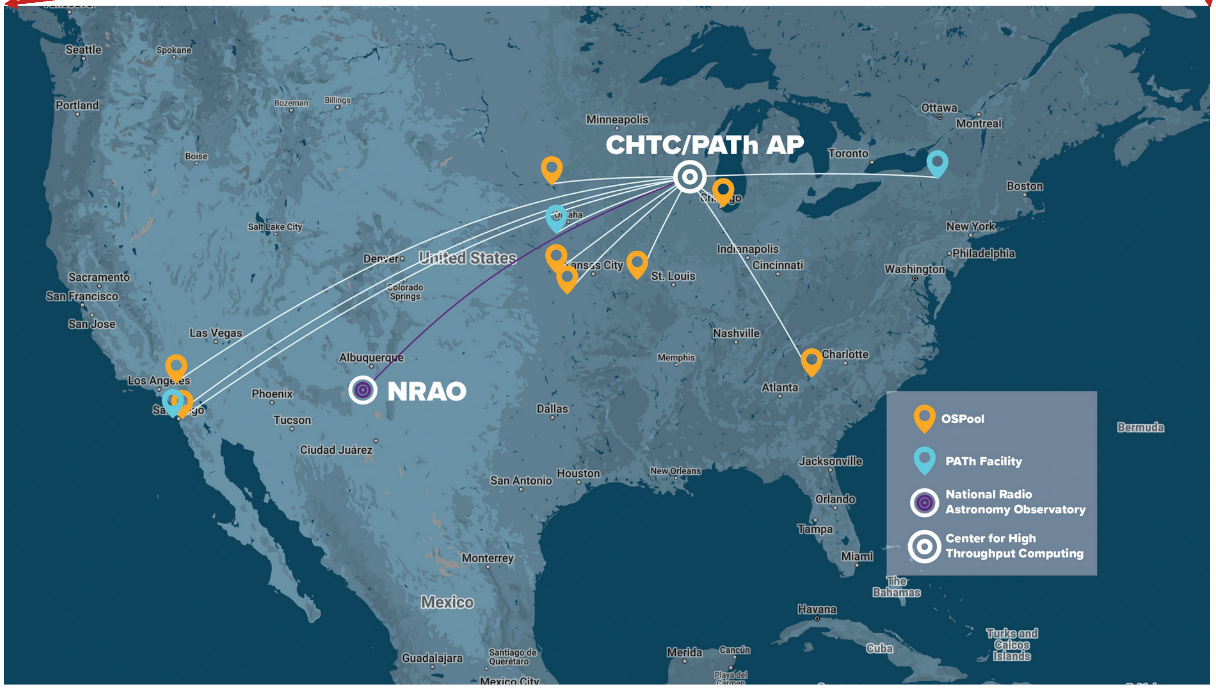


Throughput measurements

- Deployed on a cluster of GPUs (100) on the PATh facility in collaboration with https://science.nrao.edu/enews/17.3/index.shtml#deepimaging
  - The Center for High Throughput Computing (CHTC, UW-M)
  - National Research Platform (NRP) via the OSPool, Nebraska node
  - The San Diego Super Computer Center (SDSC)
  - + Multiple university computer centers across the US

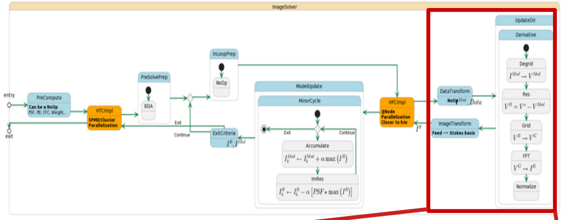
Domain specialization





Throughput measurements

- Deployed on a cluster of GPUs (100) on the PATh facility in collaboration with
  https://science.nrao.edu/enews/17.3/index.shtml#deepimaging
  - The Center for High Throughput Computing (CHTC, UW-M)
  - National Research Platform (NRP) via the OSPool, Nebraska node
  - The San Diego Super Computer Center (SDSC)
  - + Multiple university computer centers across the US



Domain specialization

- Throughput: O(1 TB/hr)
- 10 iterations in ~24 hr

- Enabling tech for many unprocessed projects in the current archive:
  - Earlier attempts using CPU cores:
    ~14 days per cycle

- This is still a small faction of the required throughput!

Throughput measurements

- Deployed on a cluster of GPUs (100) on the PATh facility
  https://science.nrao.edu/enews/17.3/index.shtml#deepimaging

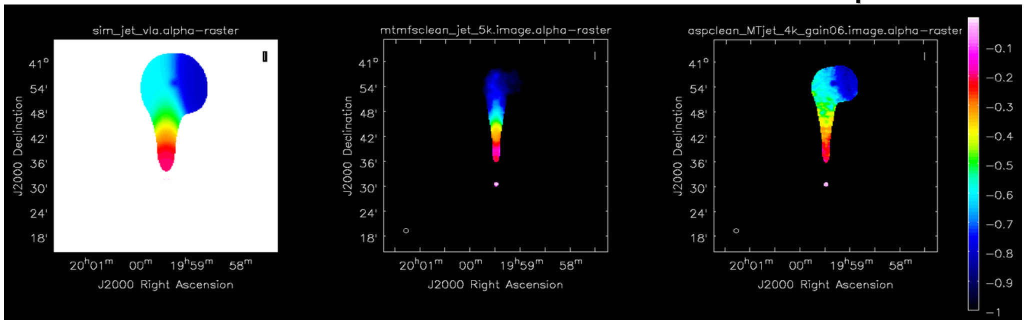
Image modeling (Model Update)

- Derivative calculations are most expensive → Design Model Update for faster convergence
- Scale-sensitive image reconstruction of complex emission
  - Asp-Clean : Narrow-band implementation (multi-algorithm modeling)
  - Wasp : Wide-band Asp
  - WiS : Wide-scale imaging (in-progress)

A&A, 426, 747-754, 2004

Spectral Index Mapping

MS-MT-MFS WAsp

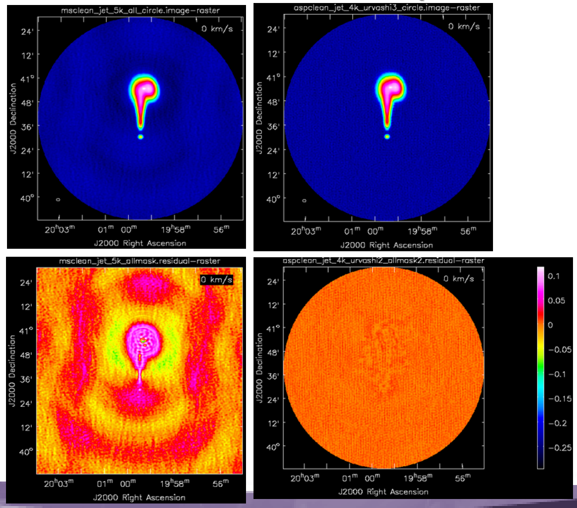


MS-MT-MFS WAsp

Stokes-I

Residual

New algorithms



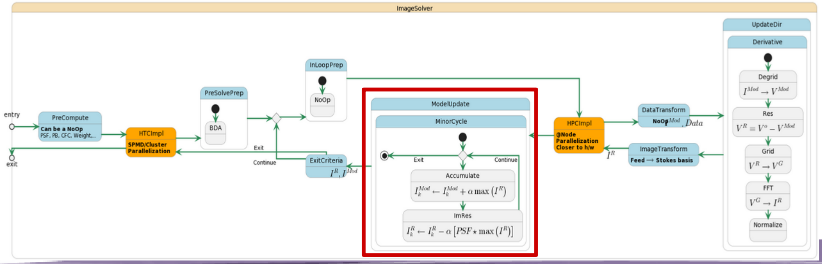


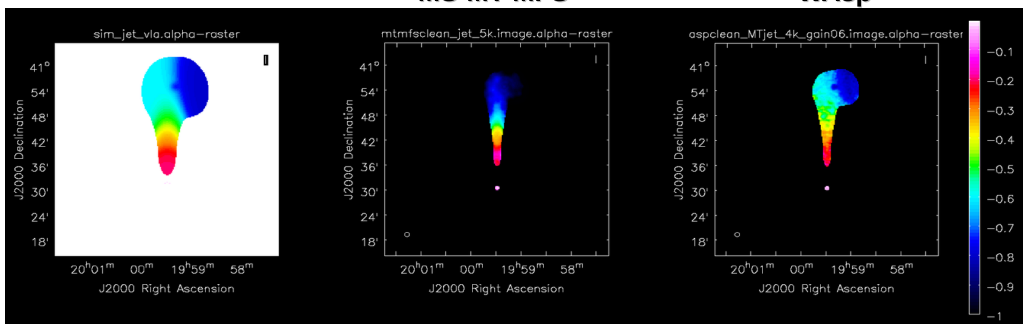
Image modeling (Model Update)

- Derivative calculations are most expensive → Design Model Update for faster convergence
- Scale-sensitive image reconstruction of complex emission
  - Asp-Clean : Narrow-band implementation (multi-algorithm modeling)
  - Wasp : Wide-band Asp
  - WiS : Wide-scale imaging (in-progress)

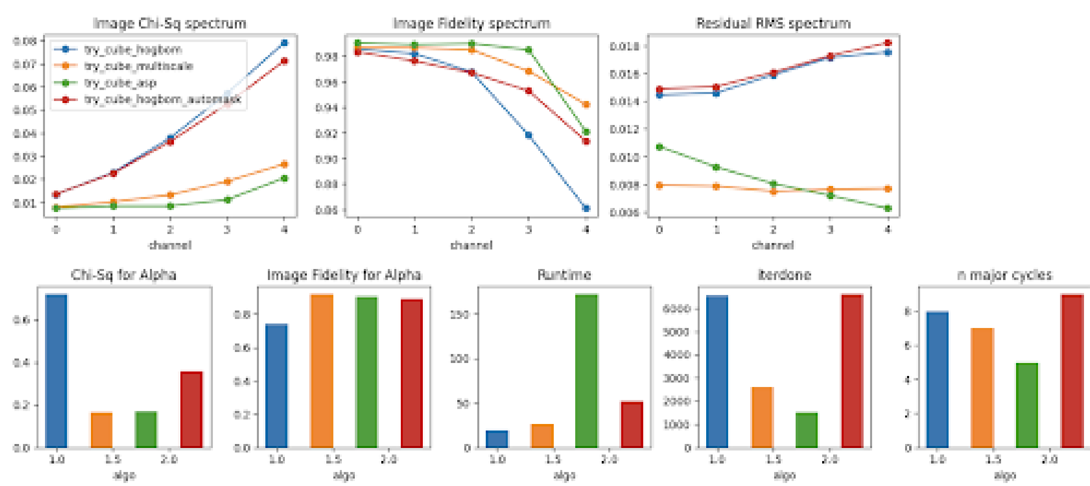
A&A, 426, 747-754, 2004

Spectral Index Mapping

MS-MT-MFS **WAsp**



New algorithms



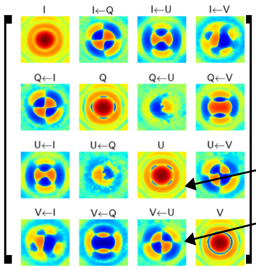
Courtesy The CASA Group

Wide-field full-Pol. Imaging

- Wide-field full polarization mapping: The concept

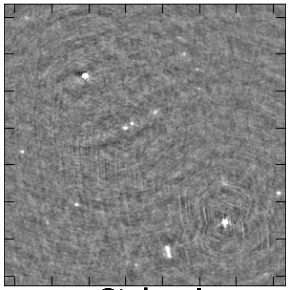
ALMA DV

$$\begin{bmatrix} I_I^{Obs} \\ I_Q^{Obs} \\ I_U^{Obs} \\ I_V^{Obs} \end{bmatrix} = \begin{bmatrix} \; \end{bmatrix} \cdot \begin{bmatrix} I_I^{o} \\ I_Q^{o} \\ I_U^{o} \\ I_V^{o} \end{bmatrix}$$

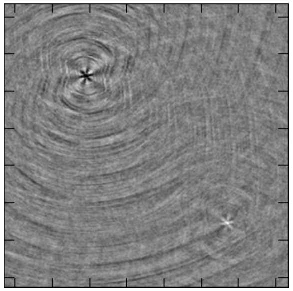


Needs holographic measurements

- Diagonal: “pure” poln. Products
- Off-diagonal: DD polarization leakage/mixing

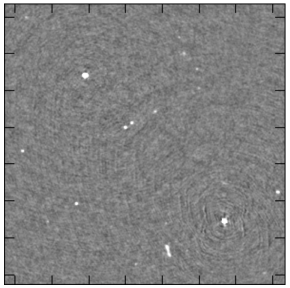


Stokes-I

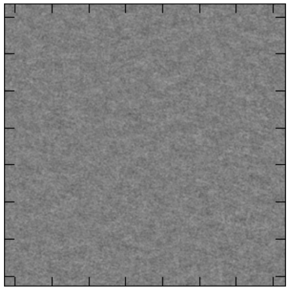


Stokes-V

Before



Stokes-I



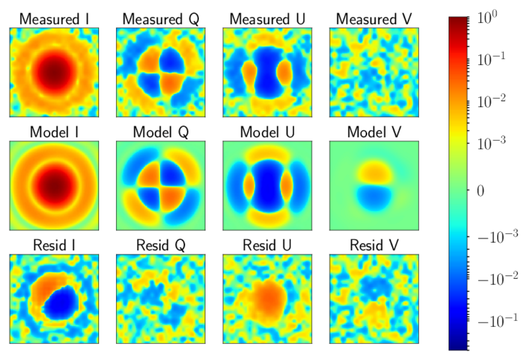
Stokes-V

After

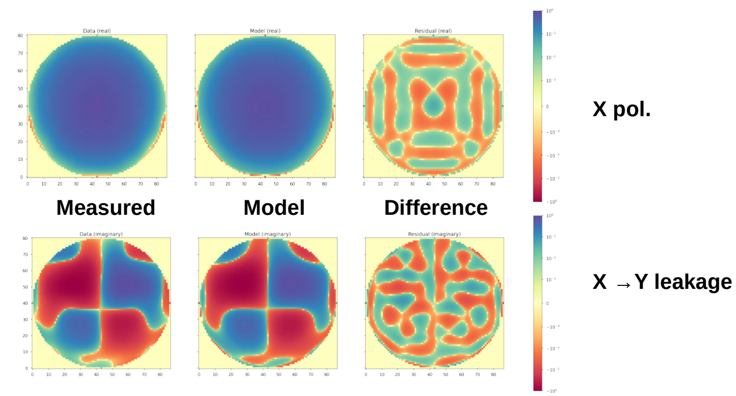
A&A 487, 419-429 (2008)

Zernike modeling for AIP (PB)

- Build a model of the antenna aperture illumination pattern (AIP)
  - Used as input to the AW-Projection framework for wide-field full-pol. imaging. Makes the algorithmic code telescope agnostic

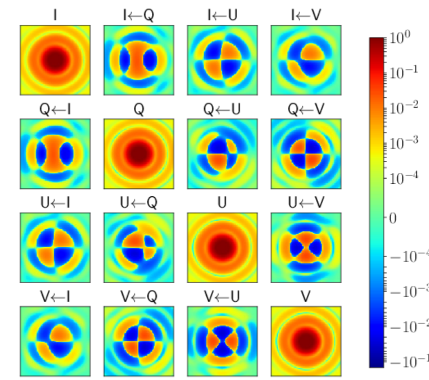


ALMA DA



MeerKAT

  - Telescope agnostic tool-chain



Holography → Plumber → *Z-coefficients* → CFCache → AWProject

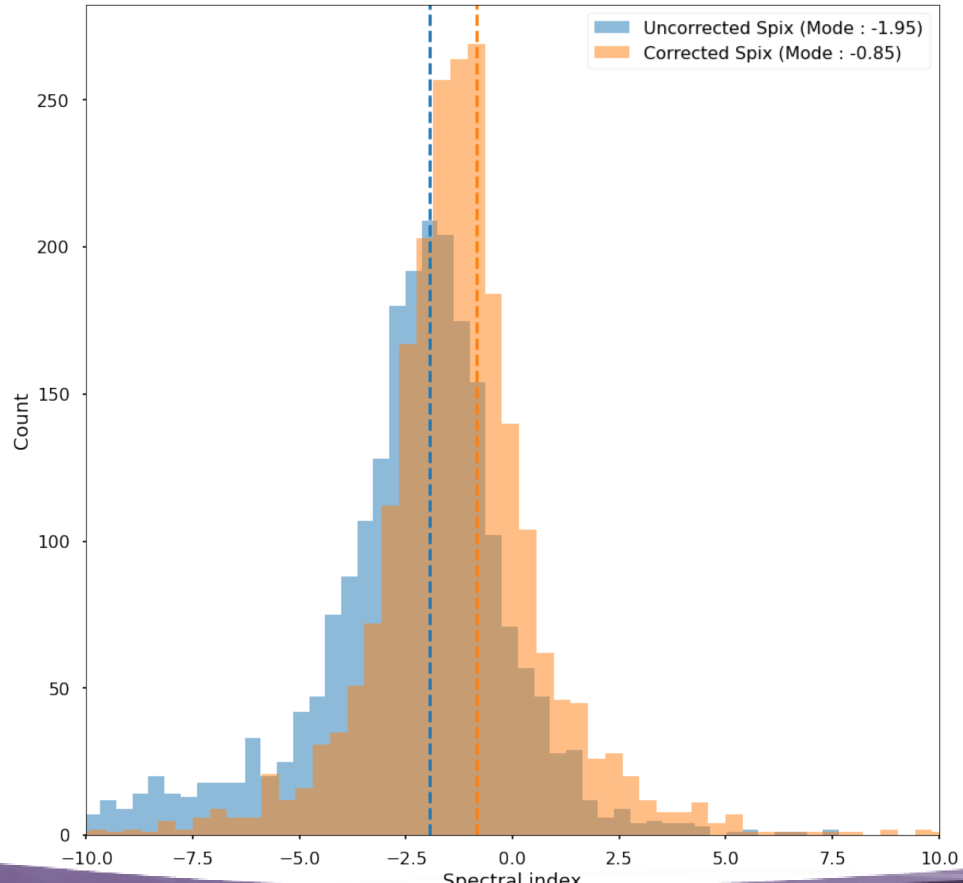
Observatory responsibility

- *Plumber* (https://github.com/ARDG-NRAO/plumber) : A general purpose package for Z-modeling of AIP, converting to PB, etc.

AJ ,163 87, 2022

Zernike modeling for AIP (PB)

- Build a model of the antenna aperture illumination pattern (AIP)
  - Used as input to the AW-Projection framework for wide-field full-pol. imaging. Makes the algorithmic code telescope agnostic



MeerKAT beam models now being used for PB correction for the MIGHTEE ~~and MALS~~ survey.

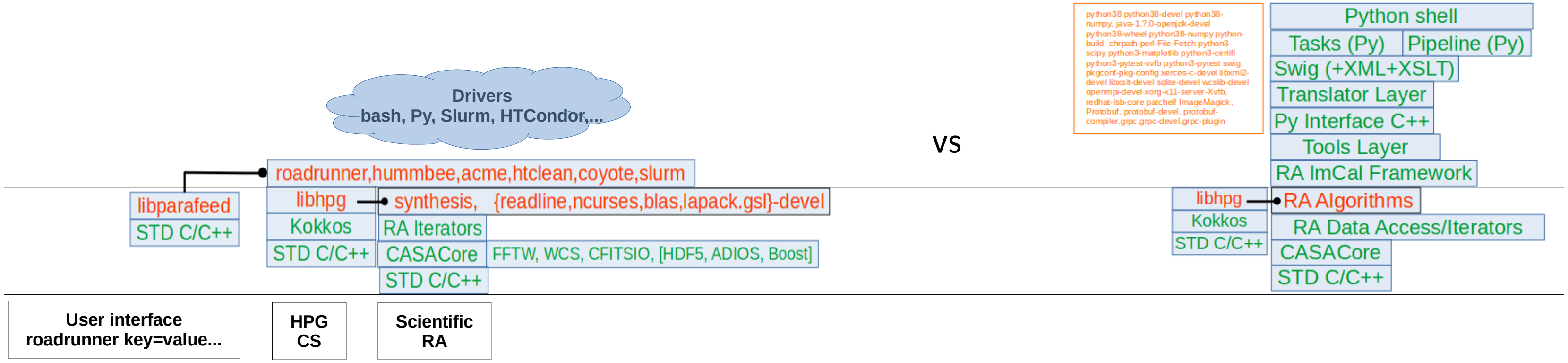
Conclusions thus far

- Scientific CASA code base is well designed, very re-usable, and reliable
  - ~~Is the scientific C++ code inherently as complex as imagined?~~ No.
  - ~~Is the entry point for new developers as hard as imagined?~~ No.
    - Successful new developers/scientists in ARDG (recall – it's a 2.5-FTE group!)
      M. (Genie) Hsieh, M. Pokorny, F. Madsen, ~~S. Sekhar,~~ H. Mueller

Conclusions thus far

- Scientific CASA code base is well designed, very re-usable, and reliable
  - ~~Is the scientific C++ code inherently as complex as imagined?~~ No.
  - ~~Is the entry point for new developers as hard as imagined?~~ No.
    - Successful new developers/scientists in ARDG (recall – it's a 2.5-FTE group!)
      M. (Genie) Hsieh, M. Pokorny, F. Madsen, ~~S. Sekhar,~~ H. Mueller

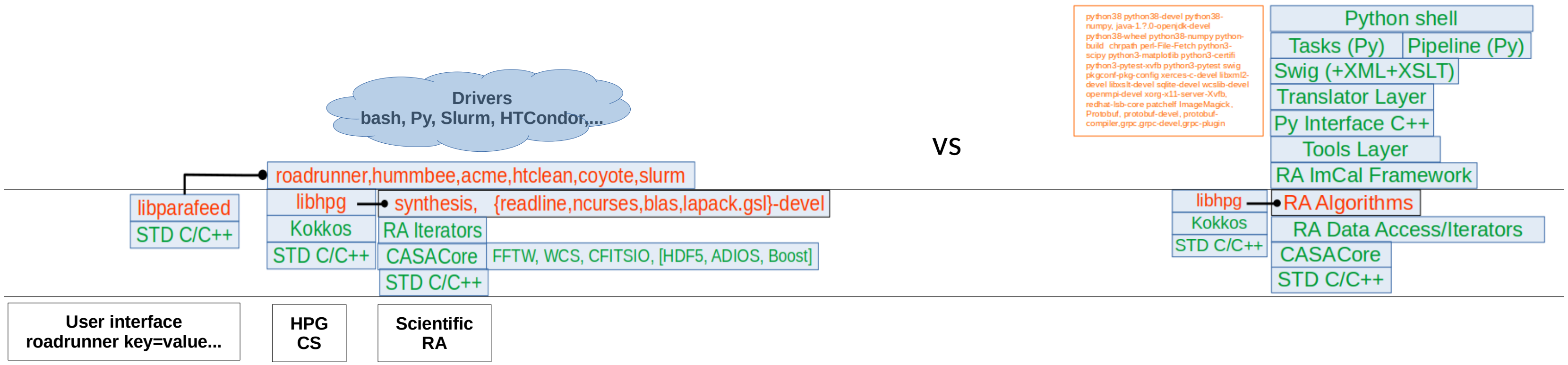
- Minimal software stack with a robust build system reduces various costs



Conclusions thus far

- Scientific CASA code base is well designed, very re-usable, and reliable
  - ~~Is the scientific C++ code inherently as complex as imagined?~~ No.
  - ~~Is the entry point for new developers as hard as imagined?~~ No.
    - Successful new developers/scientists in ARDG (recall – it's a 2.5-FTE group!)
      M. (Genie) Hsieh, M. Pokorny, F. Madsen, ~~S. Sekhar,~~ H. Mueller

- Minimal software stack with a robust build system reduces various costs

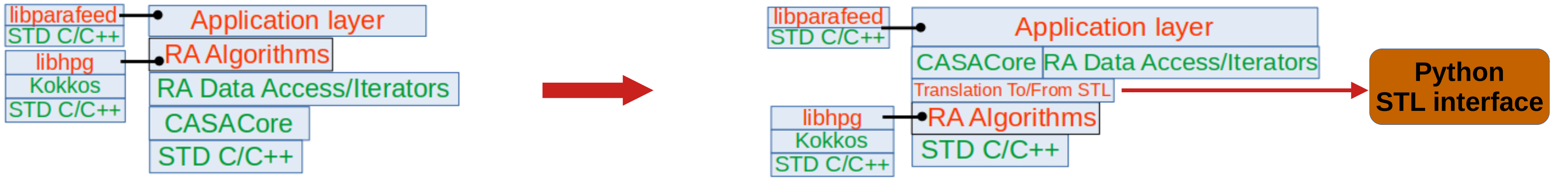


- Architectural separation of functionality, development to a design, choice of technologies, and deeper understanding, keeping real use-cases (even some users) in the loop – all are important!

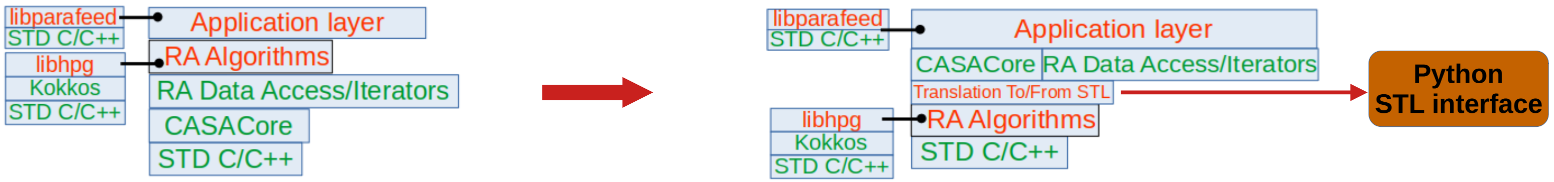
  - [Kokkos+libhpg]: HPC in a h/w independent manner
  - [CASACore+libsynthesis]: Re-use of the most advanced, highly tested RA domain scientific code-base
  - Enabling solutions: An example of rapid deployment of scientific capability

Work in progress

-

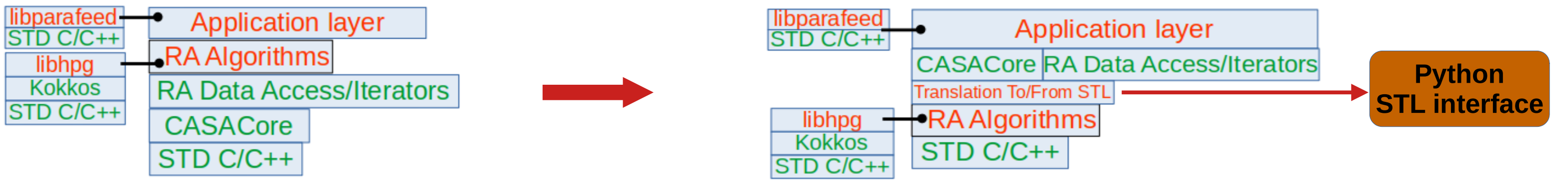


Work in progress

- 

- Performance engineering work: NVIDIA, Kokkos/SNL,...
  - Working relationships with other groups.
    - The Kokkos group: A well established HPC R&D group that developed production code.
    - CHTC: HTC group, other communities with similar computing problem (not AI!)
    - NVIDIA (new h/w), SNL, LANL,...

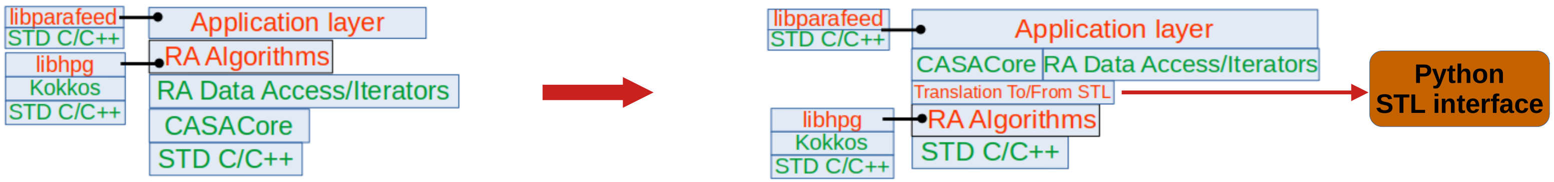
Work in progress

- 

- Performance engineering work: NVIDIA, Kokkos/SNL,...
  - Working relationships with other groups.
    - The Kokkos group: A well established HPC R&D group that developed production code.
    - CHTC: HTC group, other communities with similar computing problem (not AI!)
    - NVIDIA (new h/w), SNL, LANL,...

- Consolidate lessons from the PATh experiments
  - Getting ready for new tests + simulation + SLURM
  - Make it accessible for collaborators, other interested users

Work in progress

- 

- Performance engineering work: NVIDIA, Kokkos/SNL,…
  - Working relationships with other groups.
    - The Kokkos group: A well established HPC R&D group that developed production code.
    - CHTC: HTC group, other communities with similar computing problem (not AI!)
    - NVIDIA (new h/w), SNL, LANL,…

- Consolidate lessons from the PATh experiments
  - Getting ready for new tests + simulation + SLURM
  - Make it accessible for collaborators, other interested users

- Algorithms R&D: Wide-scale imaging
- ngVLA Simulation, algorithm verification
  - O(100TB). Storage is a bottleneck