

Usage Estimate and Computer Time Proposals

Yan-Fei Jiang (姜燕飞)

Center for Computational Astrophysics,
Flatiron Institute

Rusty and Popeye

#Nodes	CPU type	#Cores	Memory	GPU	Fabric	SLURM
216	icelake	64	1TB	no	n/a	-C icelake
640	rome	128	1TB	no	InfiniBand	-C rome,ib
300	skylake	40	768GB	no	OmniPath	-C skylake,opa
18	icelake	64	1TB	8x Nvidia H100-80GB	InfiniBand	-C h100,ib
36	icelake	64	1TB	4x Nvidia A100-80GB	InfiniBand	-C a100-80gb,ib
36	icelake	64	1TB	4x Nvidia A100-40GB	InfiniBand	-C a100,ib
22	skylake	36	768GB	4x Nvidia V100-32GB	n/a	-C v100
5	skylake	36	384GB	2x Nvidia V100-16GB	n/a	-C v100
4	cascadelake, cooperlake	96-192	3-6TB	no	n/a	-p mem

Rusty

~100K CPUs:

900M core hours per year

~500 GPUs:

4.38M hours per year

#Nodes	CPU type	#Cores	Memory	GPU	Fabric	SLURM
144	skylake	48	768GB	no	InfiniBand	-C skylake,ib
432	cascadelake	48	768GB	no	InfiniBand	-C cascadelake,ib
216	icelake	64	1TB	no	InfiniBand	-C icelake,ib
1	cooperlake	96	3TB	no	n/a	-p mem
32	skylake	40	768GB	4x NVidia V100-32GB	InfiniBand	-C v100,ib

Popeye

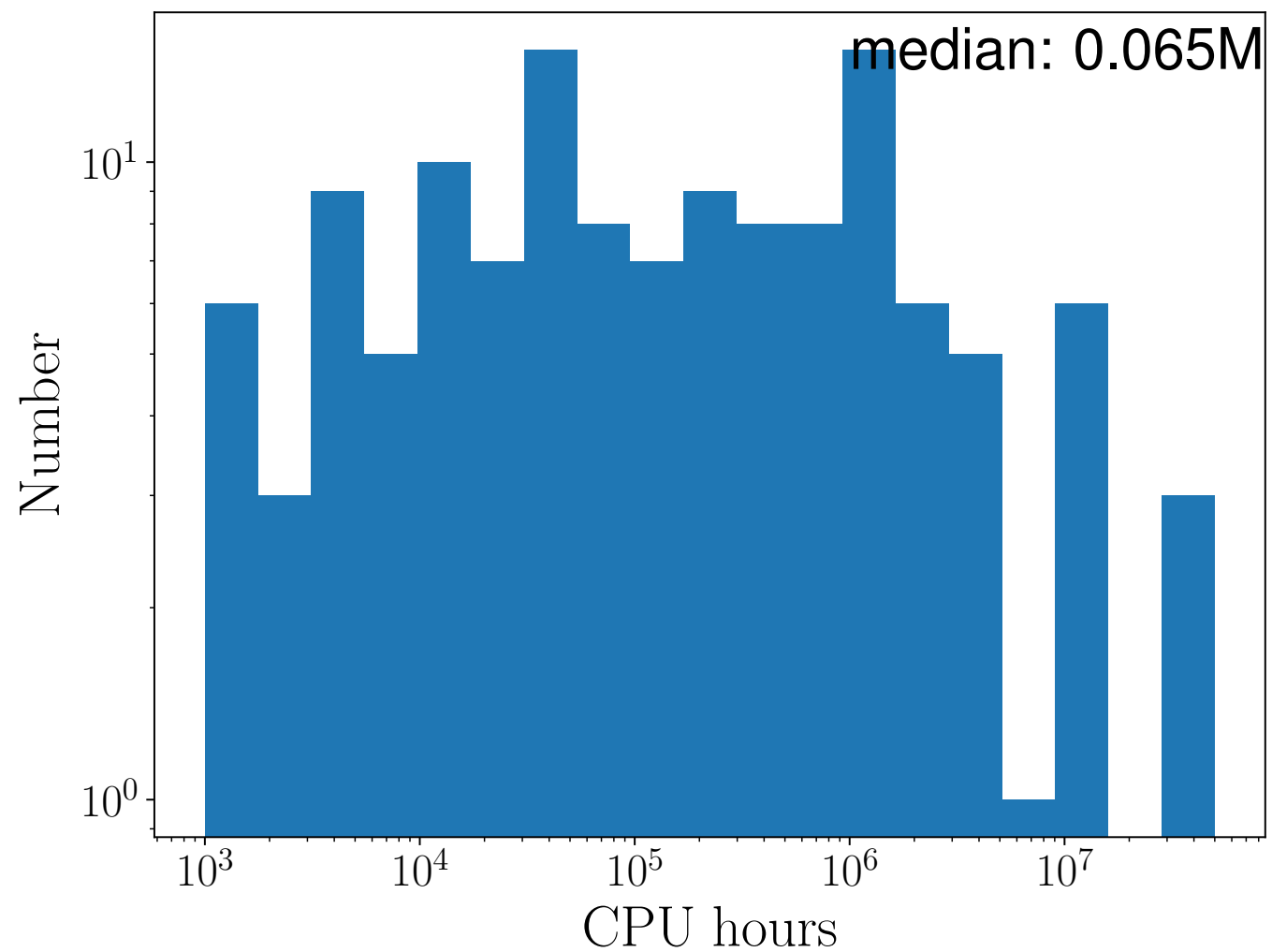
~41K CPUs:

360M core hours per year

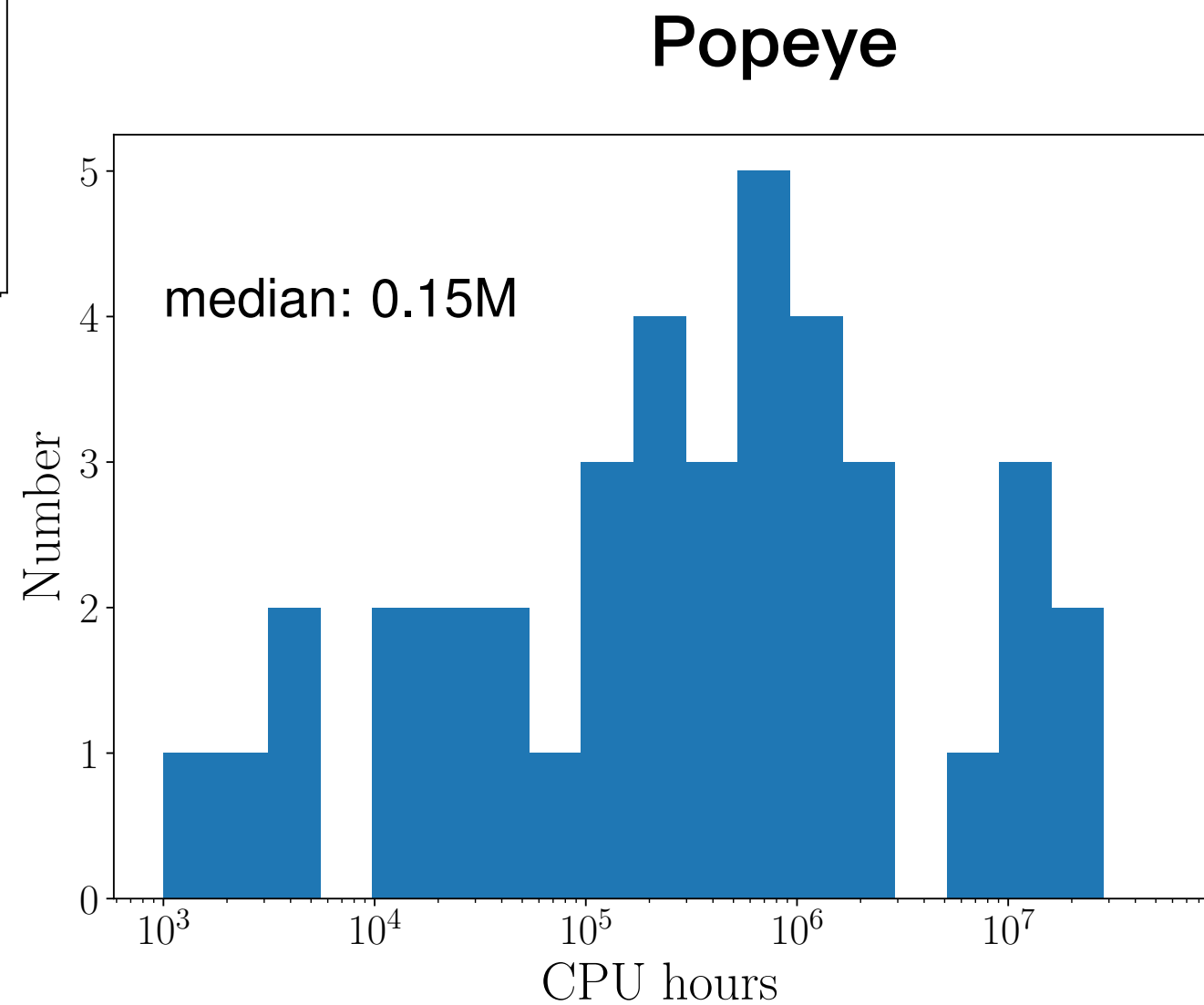
~128 GPUs:

1.12M hours per year

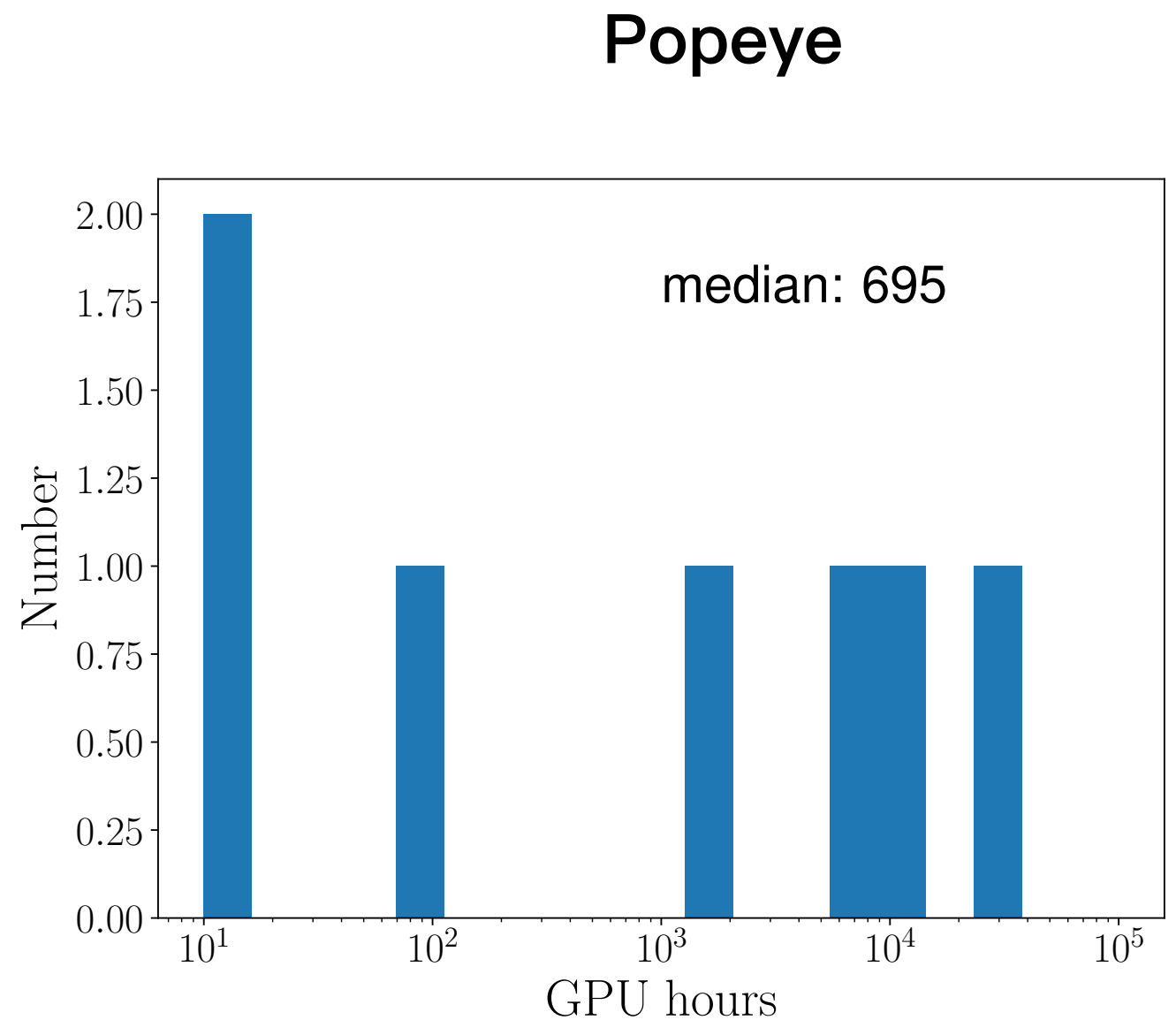
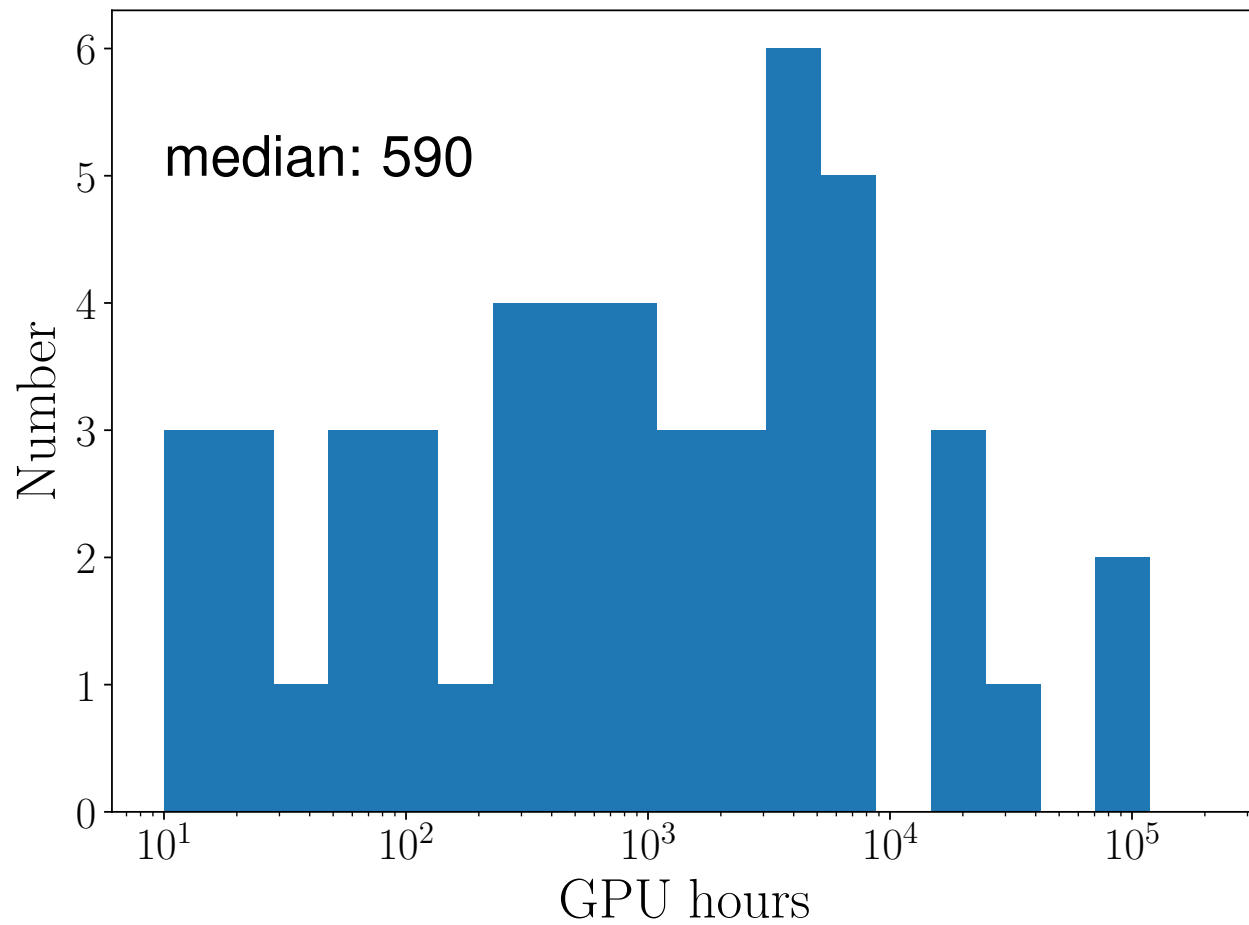
Statistics of CCA Usage



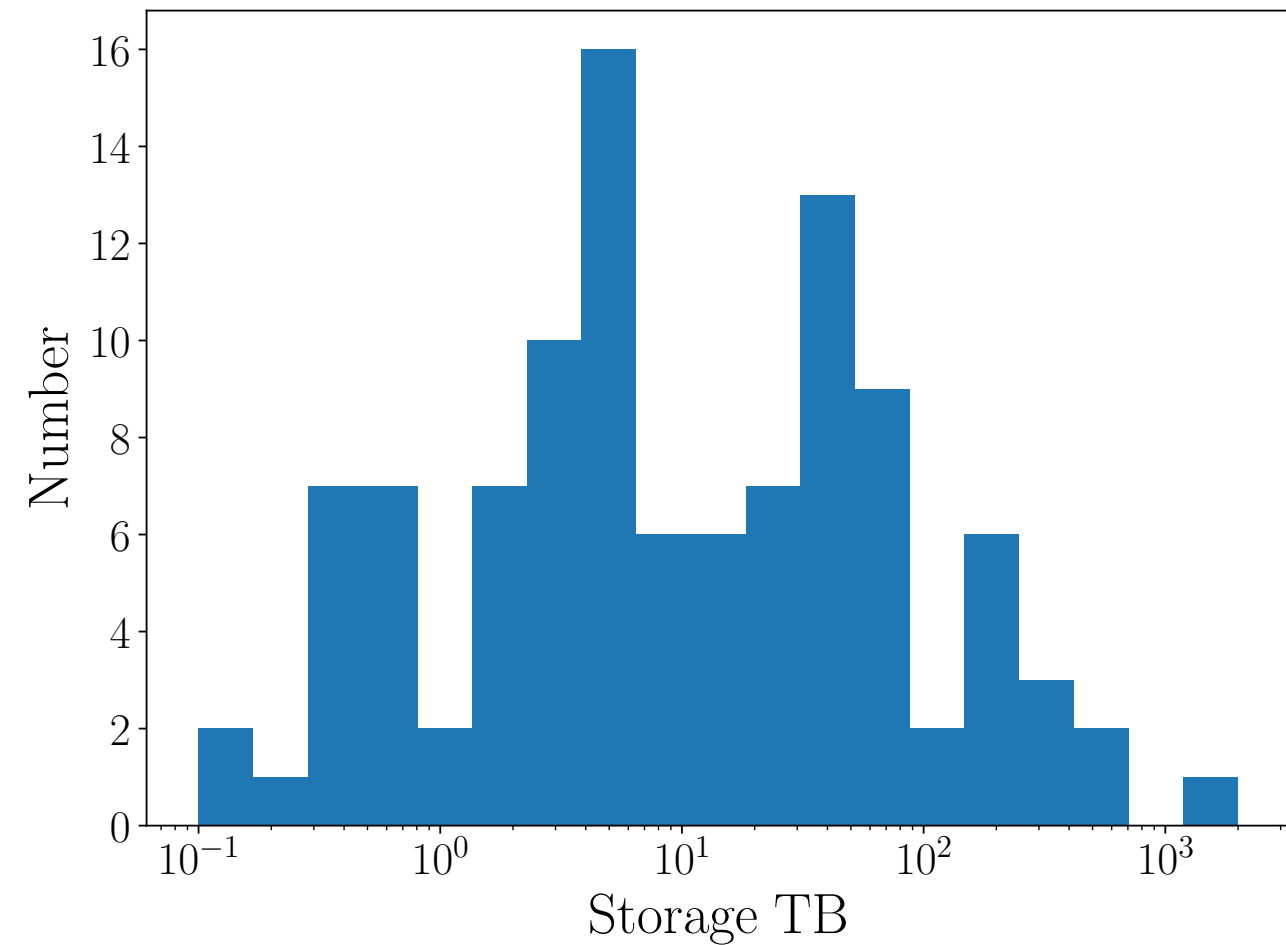
Rusty



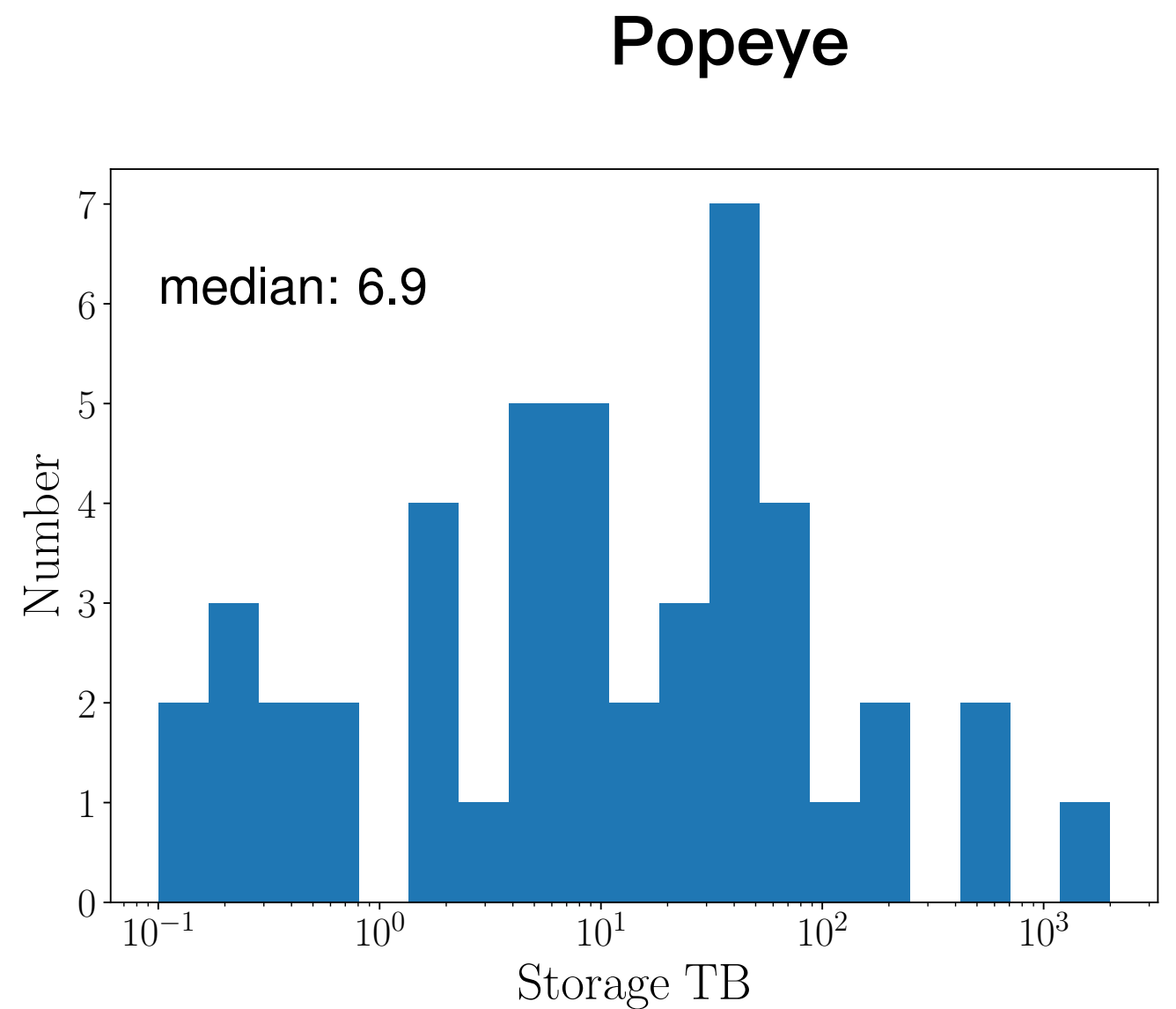
Statistics of CCA Usage



Statistics of CCA Usage



Rusty



National Supercomputers

- ACCESS
 - <https://allocations.access-ci.org/>
 - NSF supported, but no NSF grant is needed to apply
 - Can apply anytime

Allocation	Credit Threshold
<u>Explore ACCESS</u>	400,000
<u>Discover ACCESS</u>	1,500,000
<u>Accelerate ACCESS</u>	3,000,000
<u>Maximize ACCESS</u>	Not awarded in credits.

National Supercomputers

- NASA
 - <https://www.nas.nasa.gov/hecc/resources/environment.html>
 - Need NASA supported grant to get the computer time

AITKEN



Vital Stats

3,200-node HPE E-Cell/Apollo 9000 system

308,224 cores total

13.1 petaflops theoretical peak

9.1 petaflops sustained performance (April 2022)

1.27 petabytes total memory



PLEIADES



Vital Stats

10,410-node SGI/HPE ICE supercluster

228,572 cores total

7.09 petaflops theoretical peak

5.95 petaflops sustained performance (June 2021)

921 terabytes total memory



ELECTRA



Vital Stats

3,456-node SGI/HPE ICE X/HPE E-Cell system

124,416 cores total

8.32 petaflops theoretical peak

5.44 petaflops sustained performance (June 2021)

589 terabytes total memory



National Supercomputers

- INCITE
 - <https://www.alcf.anl.gov/science/incite-allocation-program>
 - DOE supported, anyone can apply.
 - Application deadline is typically in July every year

Polaris Compute Nodes

560 such
nodes in
total

POLARIS COMPUTE	DESCRIPTION	PER NODE	AGGREGATE
Processor (Note 1)	2.8 GHz 7543P	1	560
Cores/Threads	AMD Zen 3 (Milan)	32/64	17,920/35,840
RAM (Note 2)	DDR4	512 GiB	280 TiB
GPUS	NVIDIA A100	4	2240
Local SSD	1.6 TB	2/3.2 TB	1120/1.8PB

National Supercomputers

- INCITE

- <https://www.alcf.anl.gov/science/incite-allocation-program>
- DOE supported, anyone can apply.
- Application deadline is typically in July every year

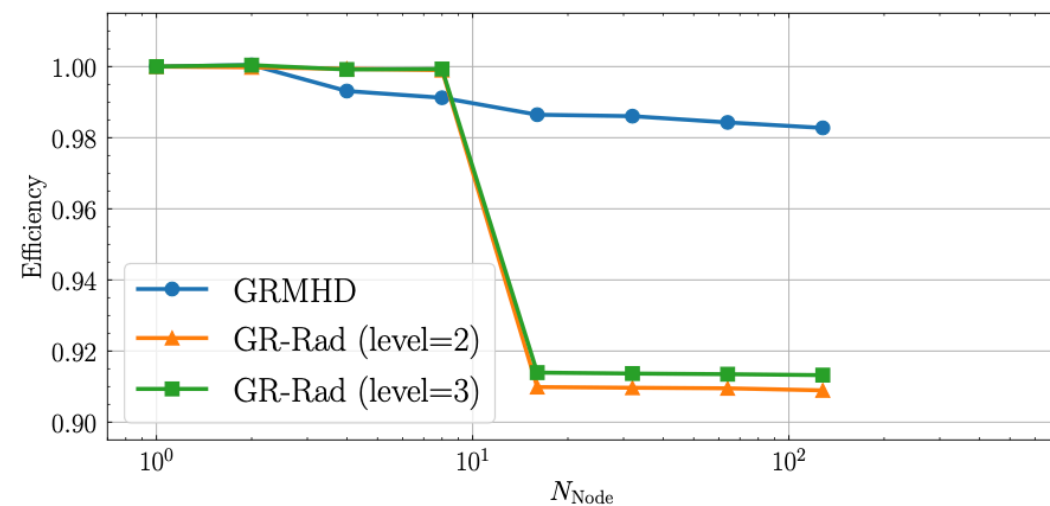
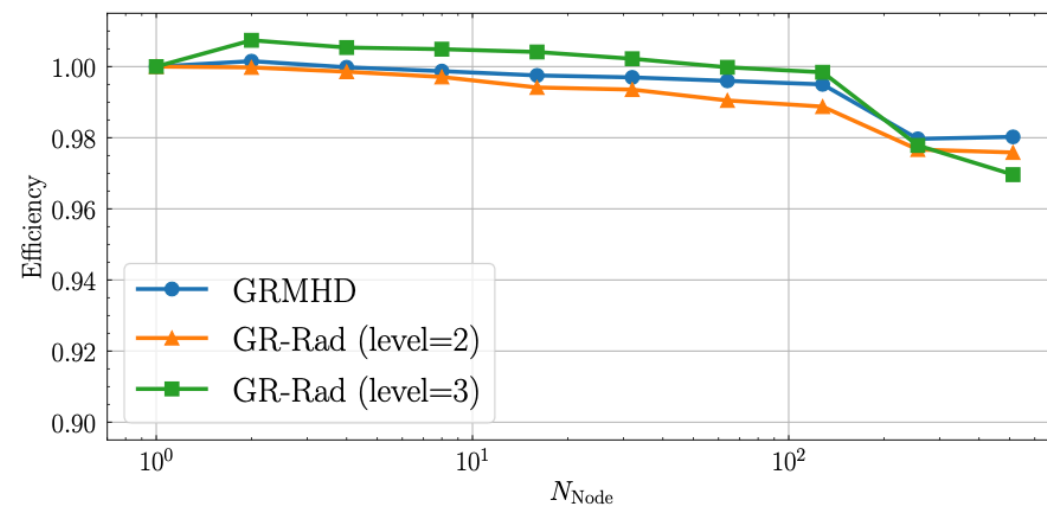
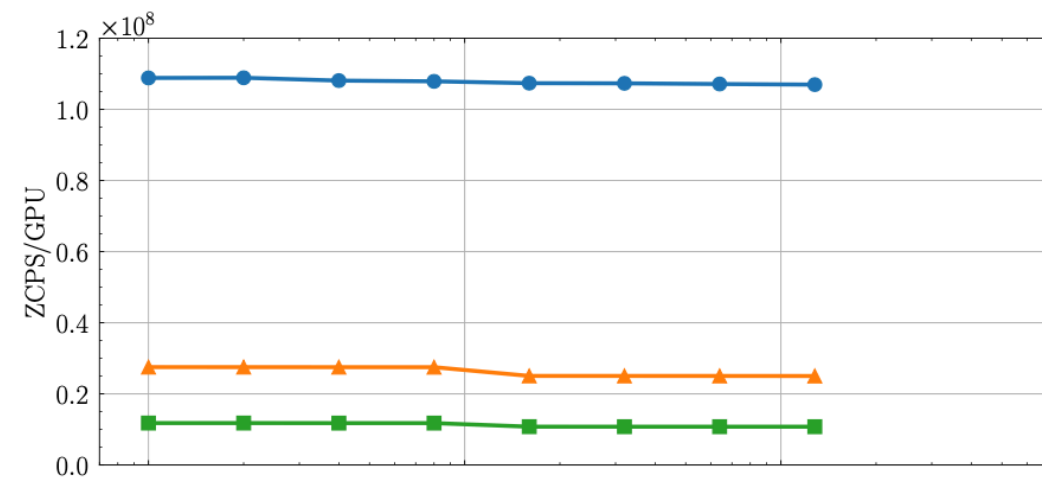
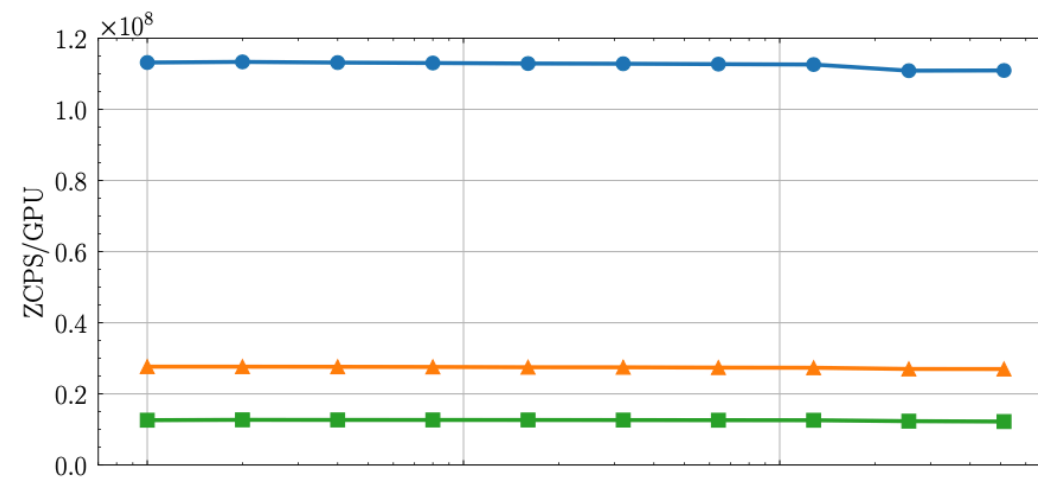
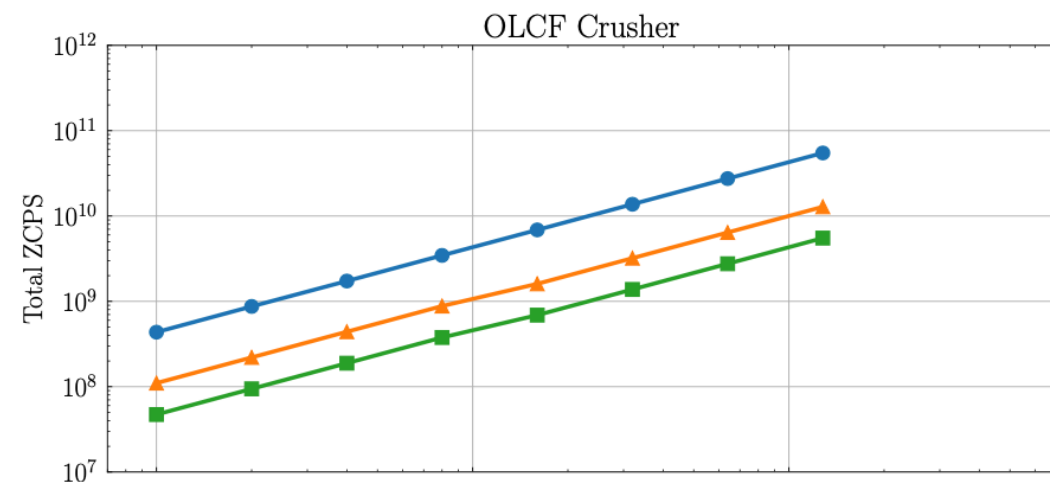
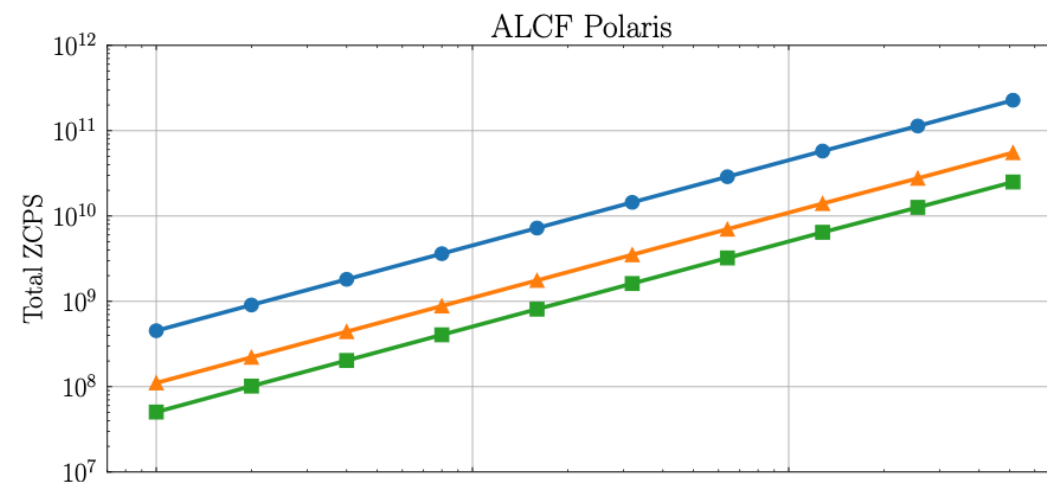
Frontier:

The system has 9408 AMD compute nodes. Each node has 1 AMD “Optimized 3rd Gen EPYC” CPU and 4x AMD MI250X, each with 2 GPUs (so total 8 GPUs per node).

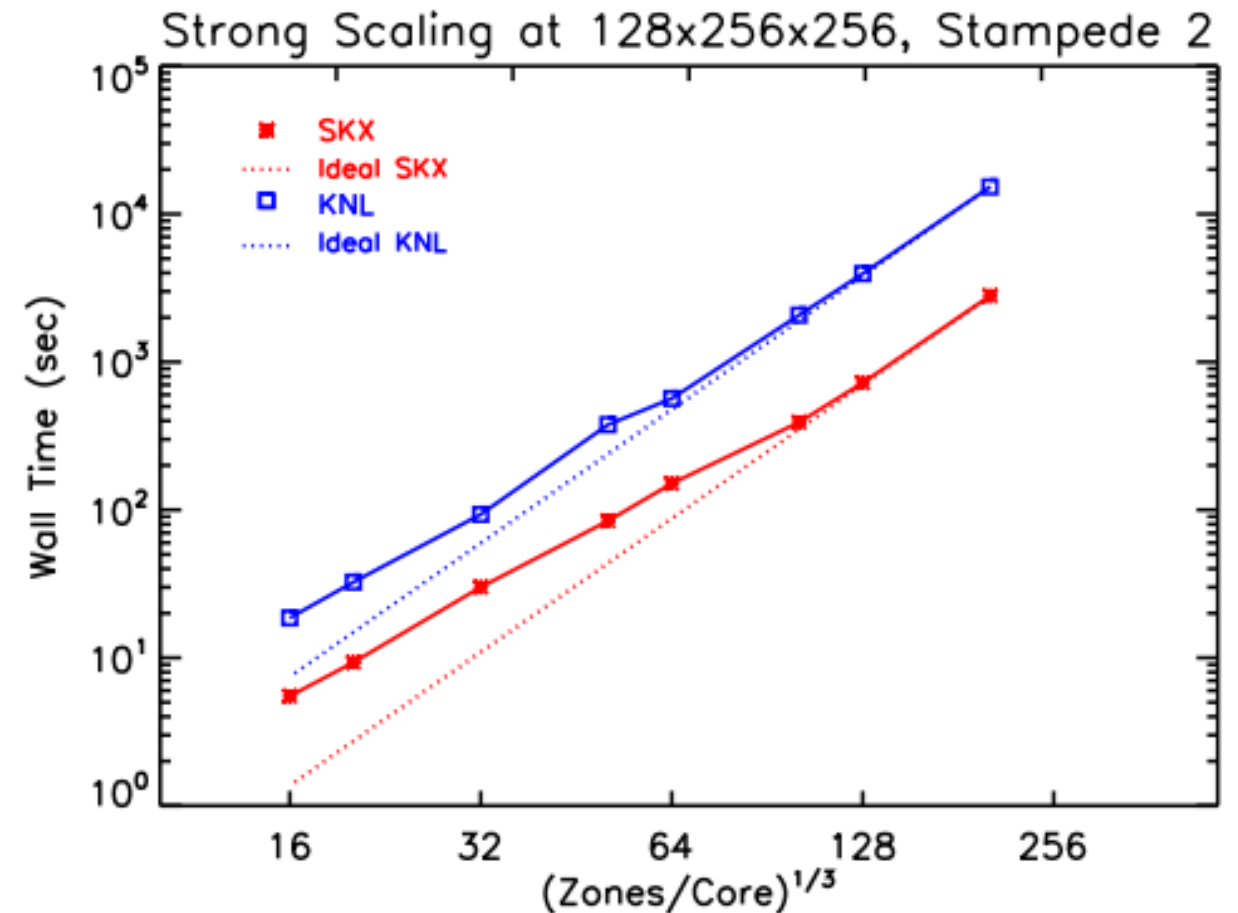
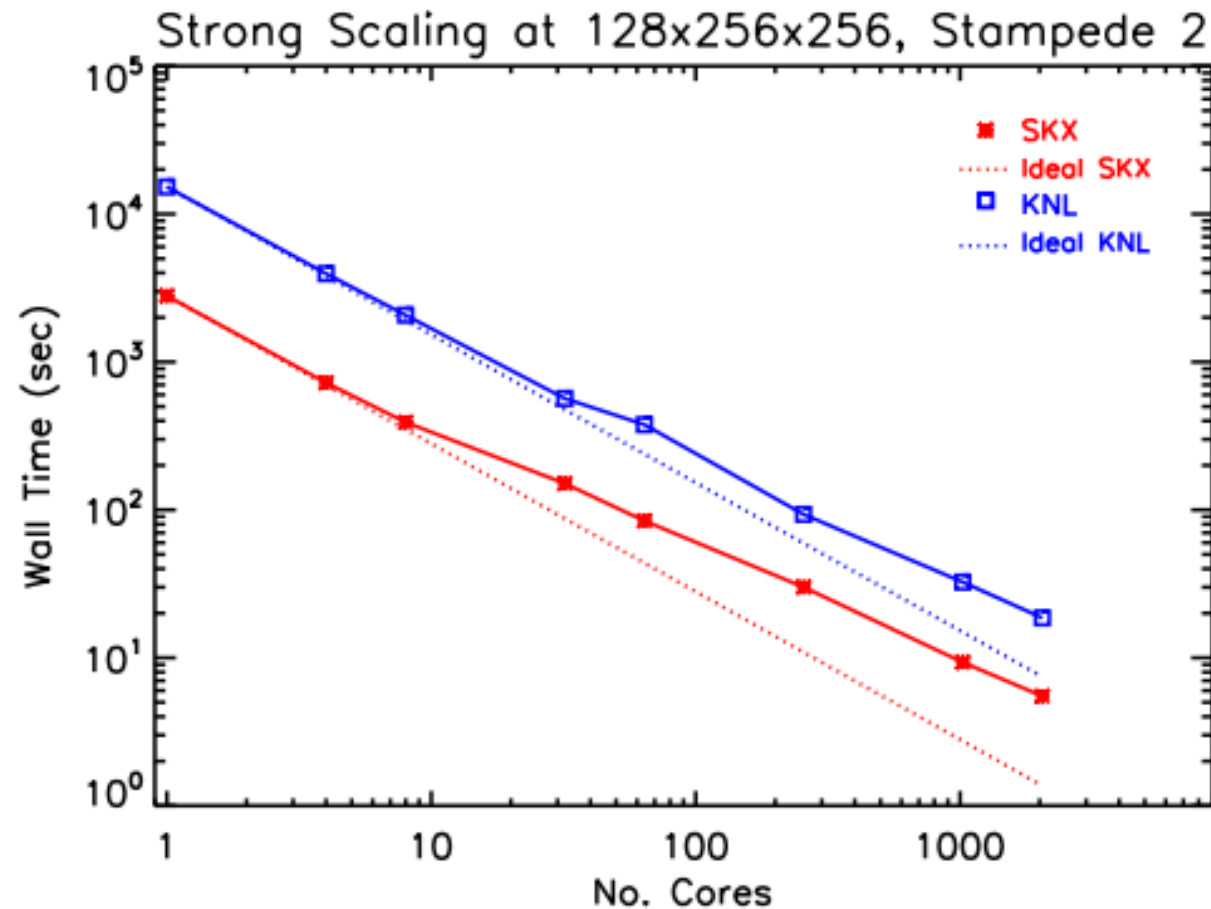
Scaling Tests

- **Strong scaling:**
For a fixed simulation you try to run, how the wall clock time changes as a function of number of cores you use
- **Weak scaling:**
For a fixed problem size per core, how the simulation efficiency varies as a function of total number of cores.
- You can typically request a small amount of allocation to perform these tests in the system you want to use with your code.

Examples of Scaling Tests



Examples of Scaling Tests



<https://allocations.access-ci.org/example-proposals/AstronomicSciences.pdf>

Estimate the Resource You need

- Total number of calculations per cycle: N
- Total number of cycles you need to do: T
- Number of calculations your code can perform per second per core: S

Total core time: $N \cdot T / S$