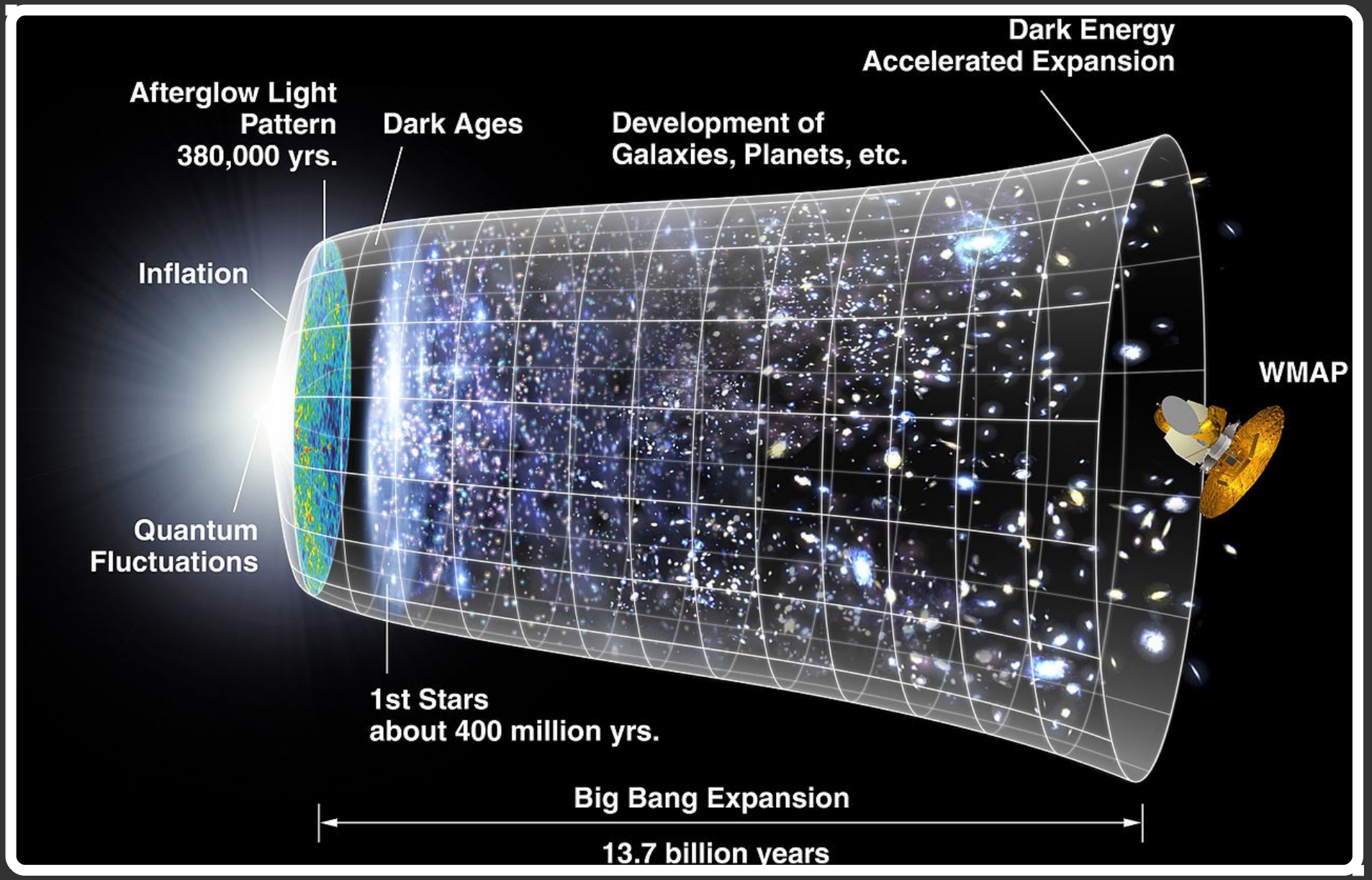
# Embedded Supercomputing: Radio Astronomy at the Limit

Simon Ratcliffe & Bruce Merry

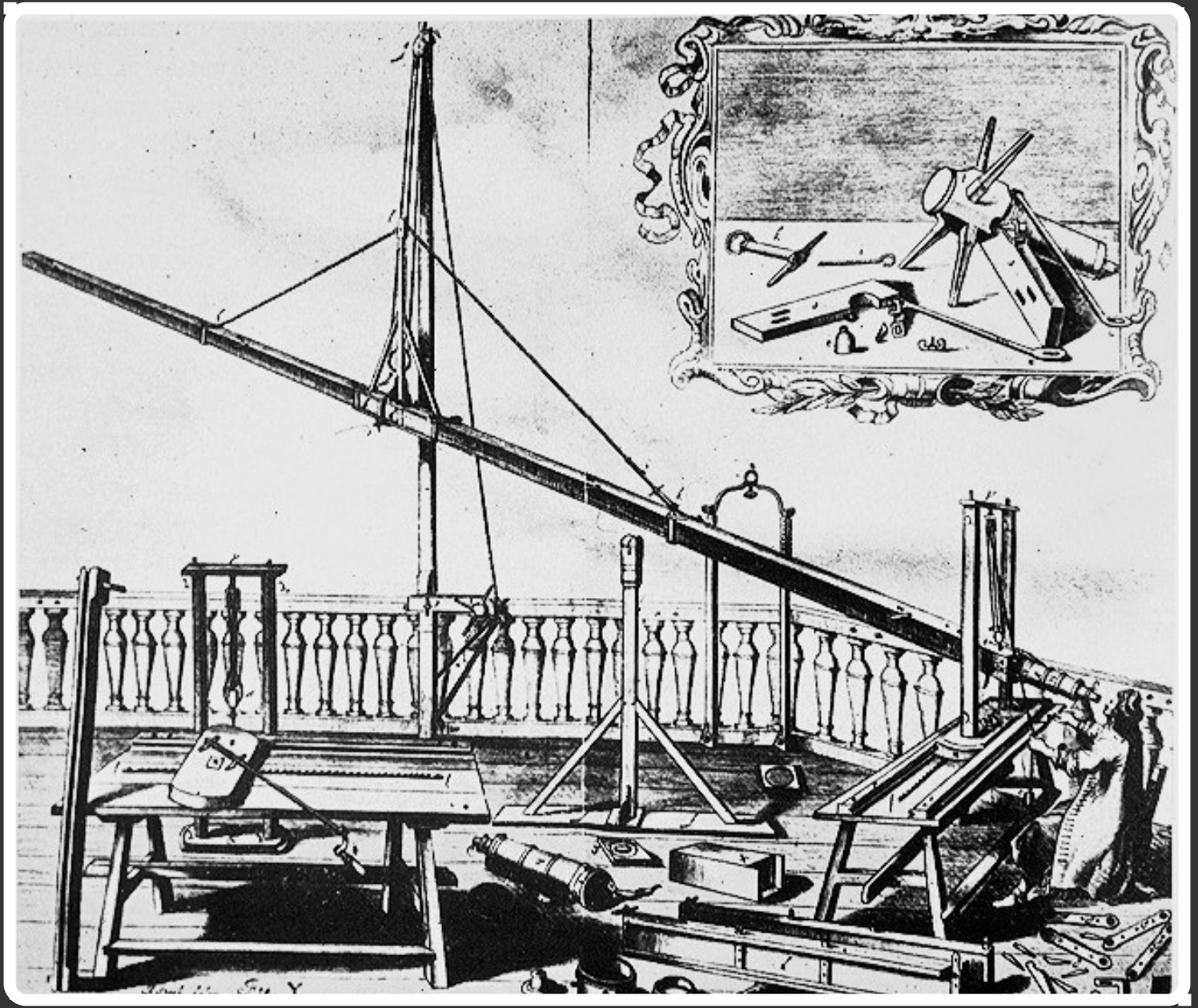
SKA South Africa

JSNTB

# In the beginning....

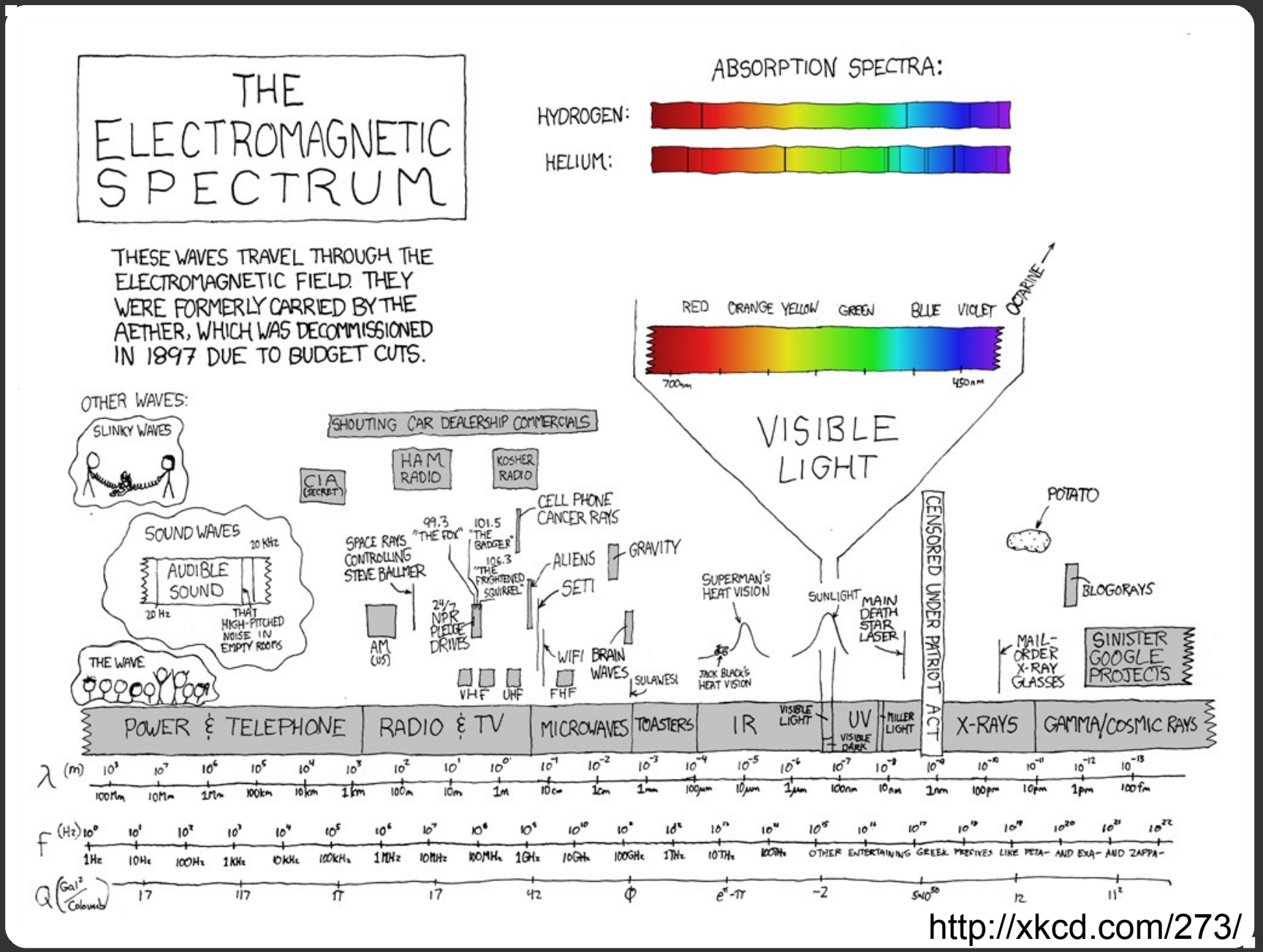


# more recently



Johannes Hevelius 60ft - 1673
(the Ewan McTeagle of his day)

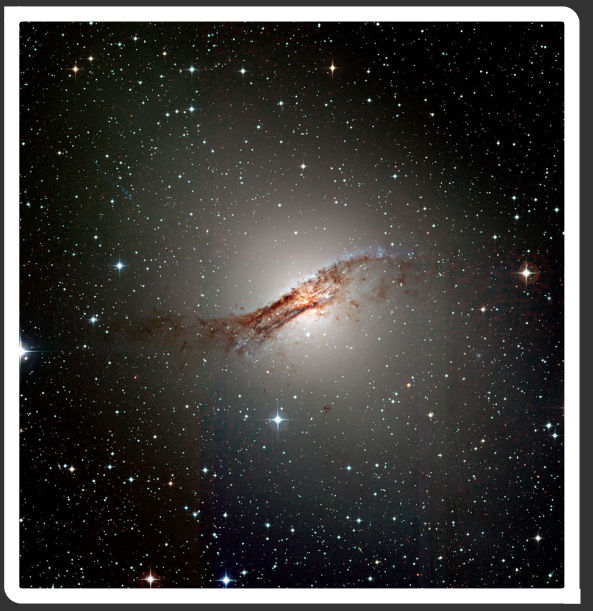
# everything we know so far




http://xkcd.com/273/

# getting the full picture

=



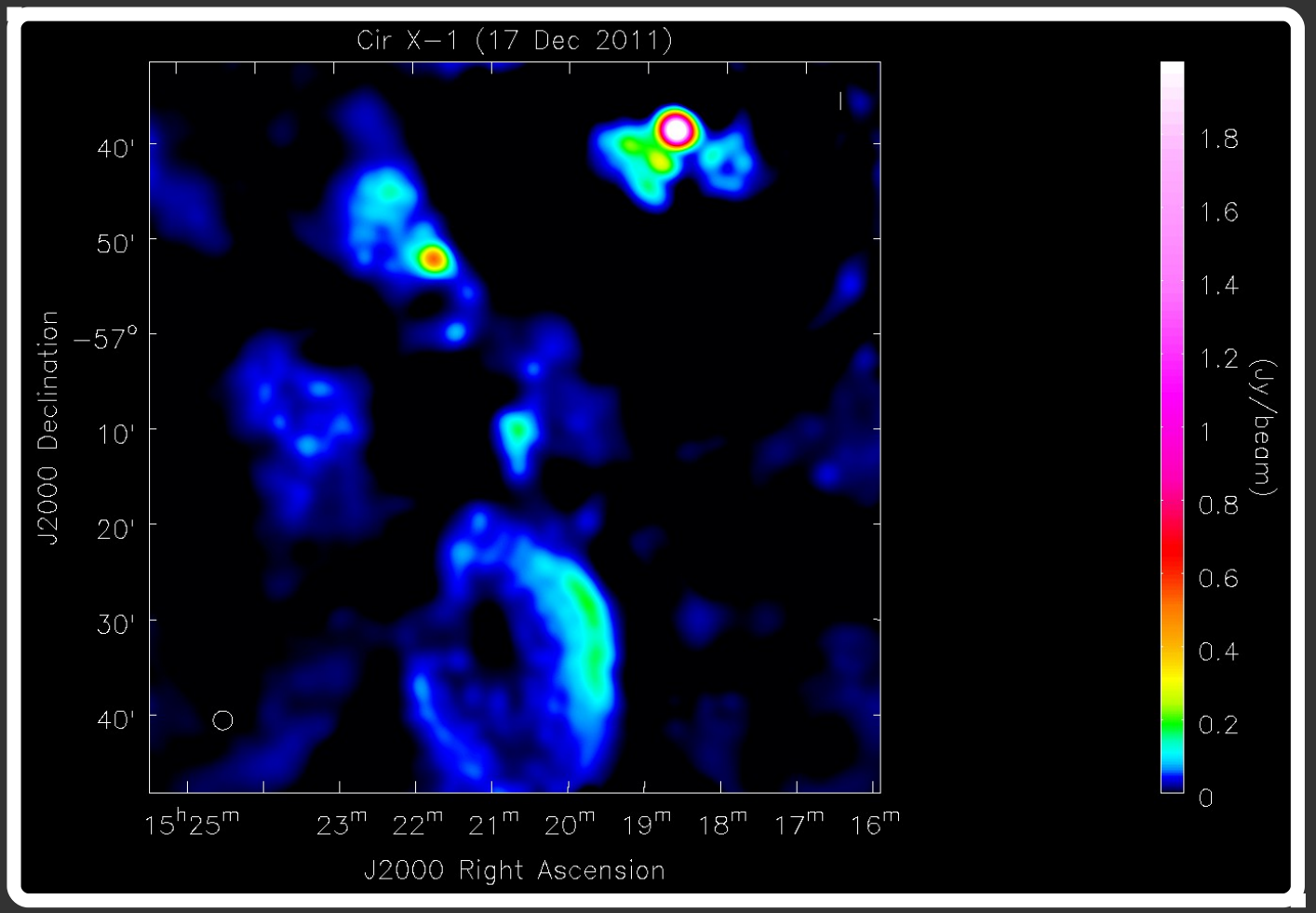
+

Jansky VLA, New Mexico

=

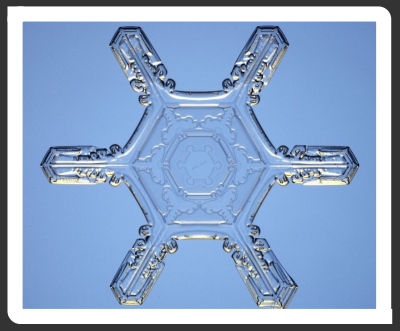
# Watt?

$$1\ \mathrm{Jy} = 10^{-26}\ \mathrm{W m^{-2} Hz^{-1}}$$




I cannae do it, captain, ye cannae change the laws of physics

# how much wood could a wood chuck, chuck

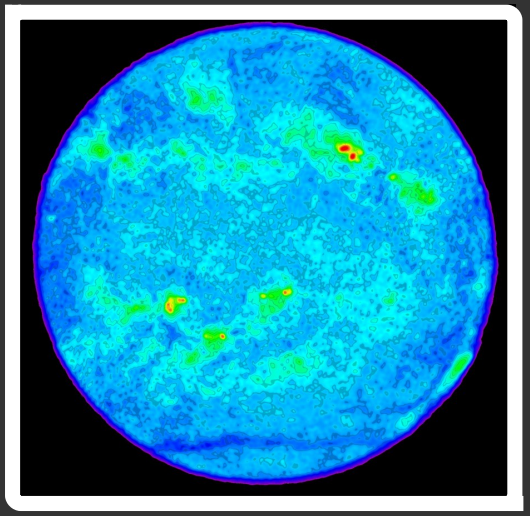


1.008 J

0.00000016 J

0.000000000000007 J

# “Know your enemy and know yourself”



$10^5$ Jy

Sun @ 5 GHz

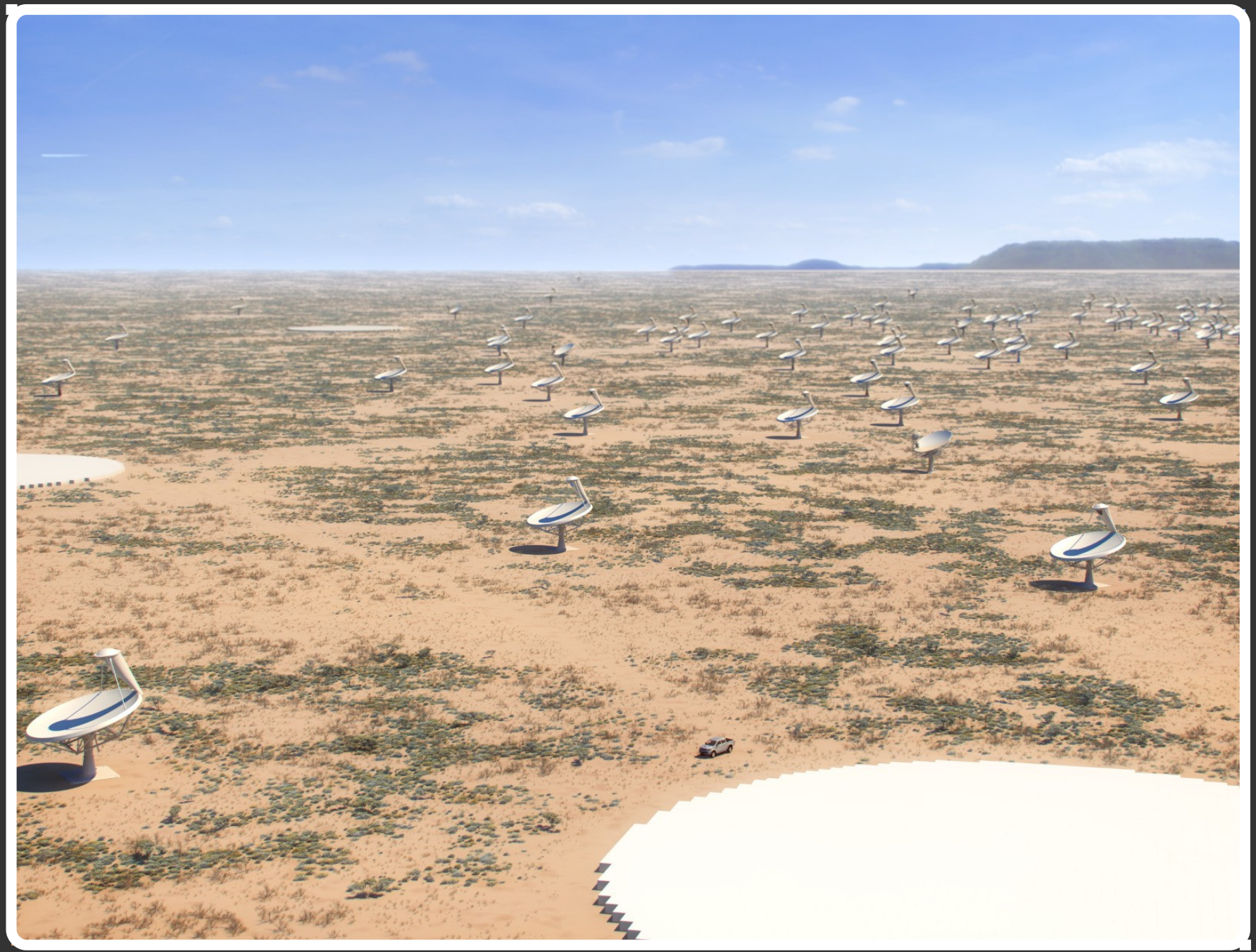

$10^8$ Jy

GSM Phone @ 1km

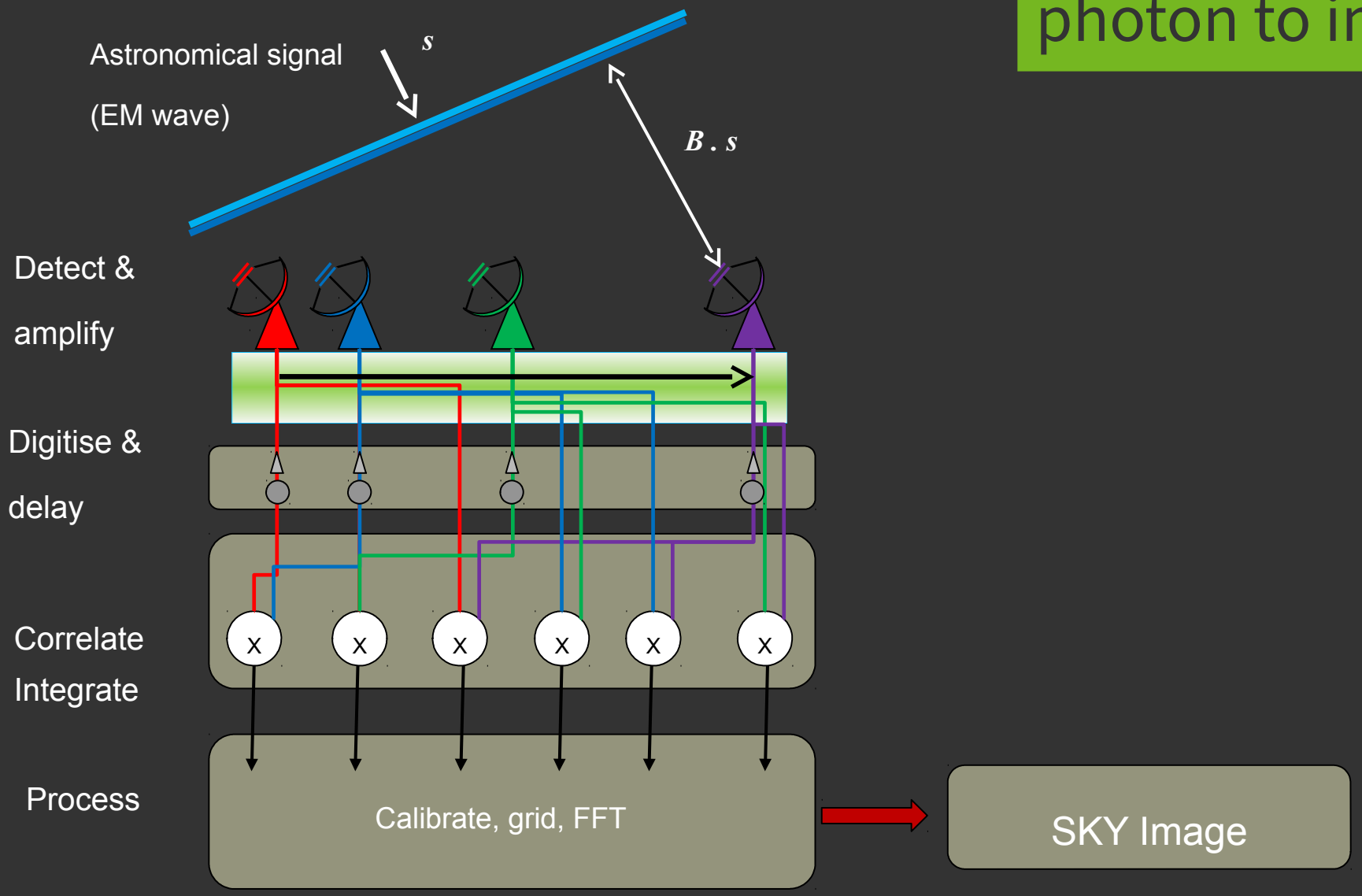
0 Jy

'Smart' Phone @ 1m

# something a little bigger



# photon to image




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

# Will it blend ?


X


# 10000000000000000000 B

@ 1 bit per grain of standardised* sand

* assumptions apply (10gpmm^3, 3kmx2kmx700m, only valid when calculated on paper napkin, just say no to assumptions)

# You sir, are a blaggard and a coward

1 exbibyte

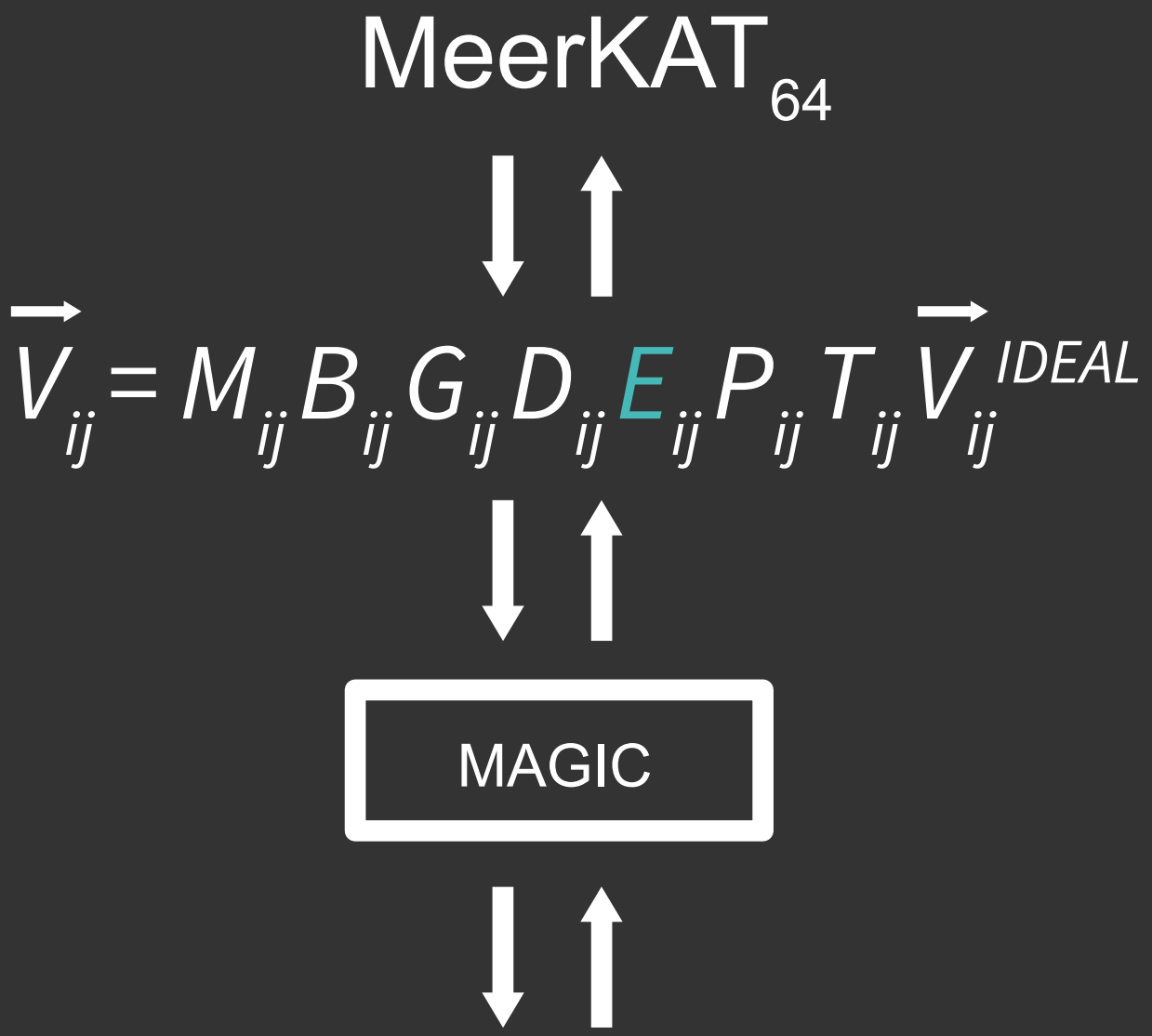
- 1 exabyte

38, 230 x

# Big Iron

# Medium Iron (and a fair bit of aluminium)

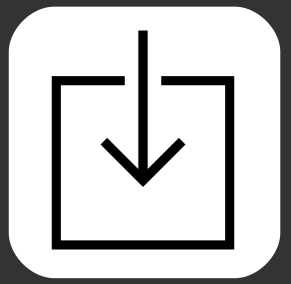
# In theory


MeerKAT64
$\vec{V}_{ij} = M_{ij} B_{ij} G_{ij} D_{ij} E_{ij} P_{ij} T_{ij} \vec{V}_{ij}^{IDEAL}$
MAGIC
IO / Cache / FLOPS / kW / $

# “My god it's full of data”

$$50^{\text{hour}}_{\text{observation}} \;@\; 68^{\text{Gibps}}_{\text{input data rate}} \;=\; 1^{\text{PiB}}_{\text{buffer}}$$

# 50 hour totals



1.1 Tibps
buffer read rate

1.7 ExaFlop
total FP operations

1.9 TB
working memory

# easy…..

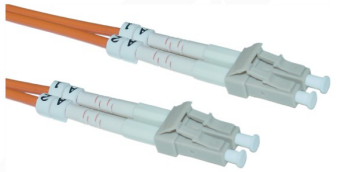
# whither Mr. Fusion ?

20 kW

available power

# rosetta stone (edition 2010)

Immersive cooling
No fans
Glycol ground loop

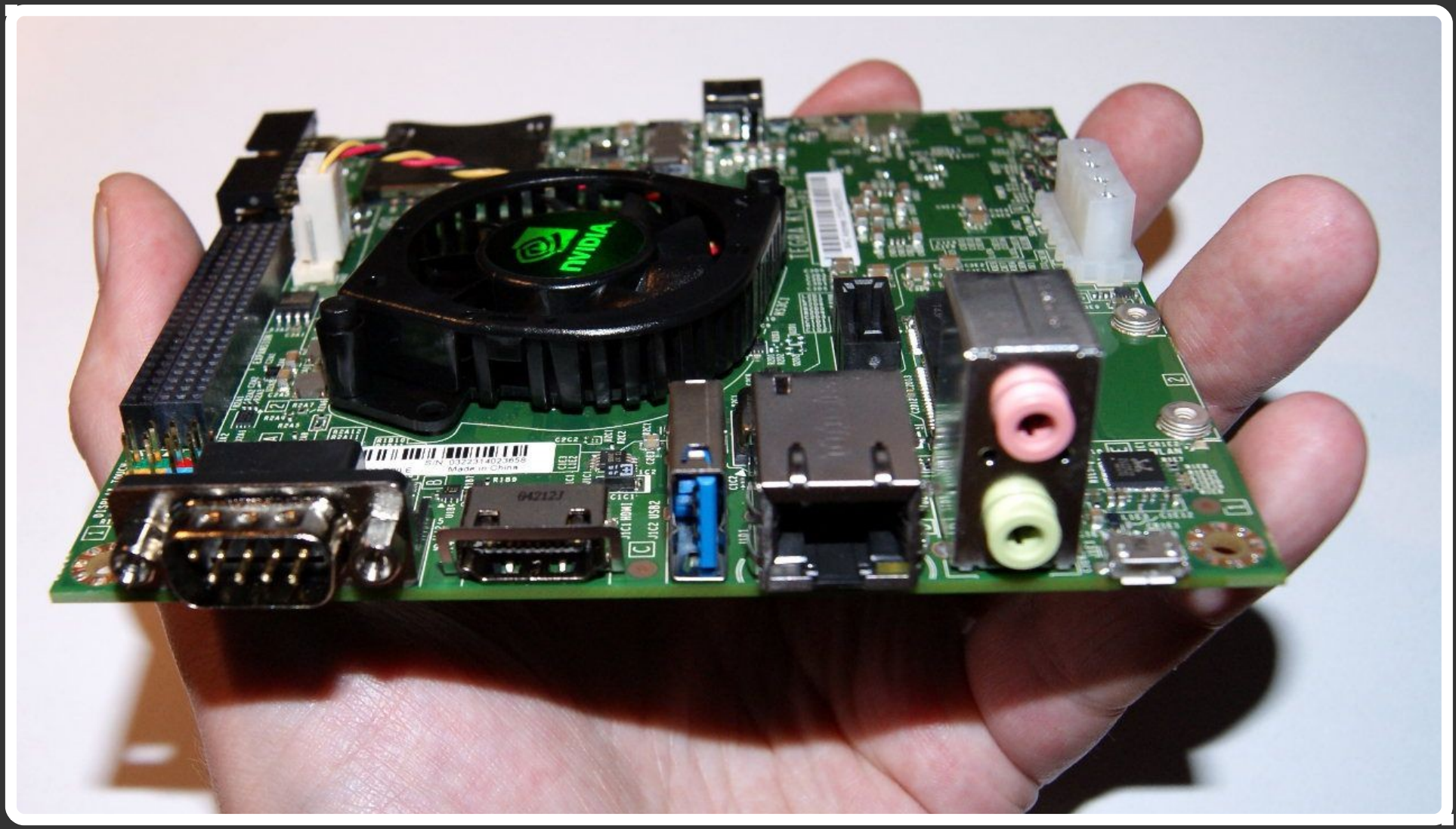


20 GBps In
2000 nodes
100 Mbpn


Tegra SoC
1W TDP
Cuda GPU
SO DIMM

50 TFlops, 2 TB RAM, 8 TB Flash
2.5 kW Power, 0 kW Cooling, $350k built

# ...shake your windows, and rattle your walls




TK1: 327 GFLOPs / 12.7 GiBps

# measurement equation (redux)

baseline

direction

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

input

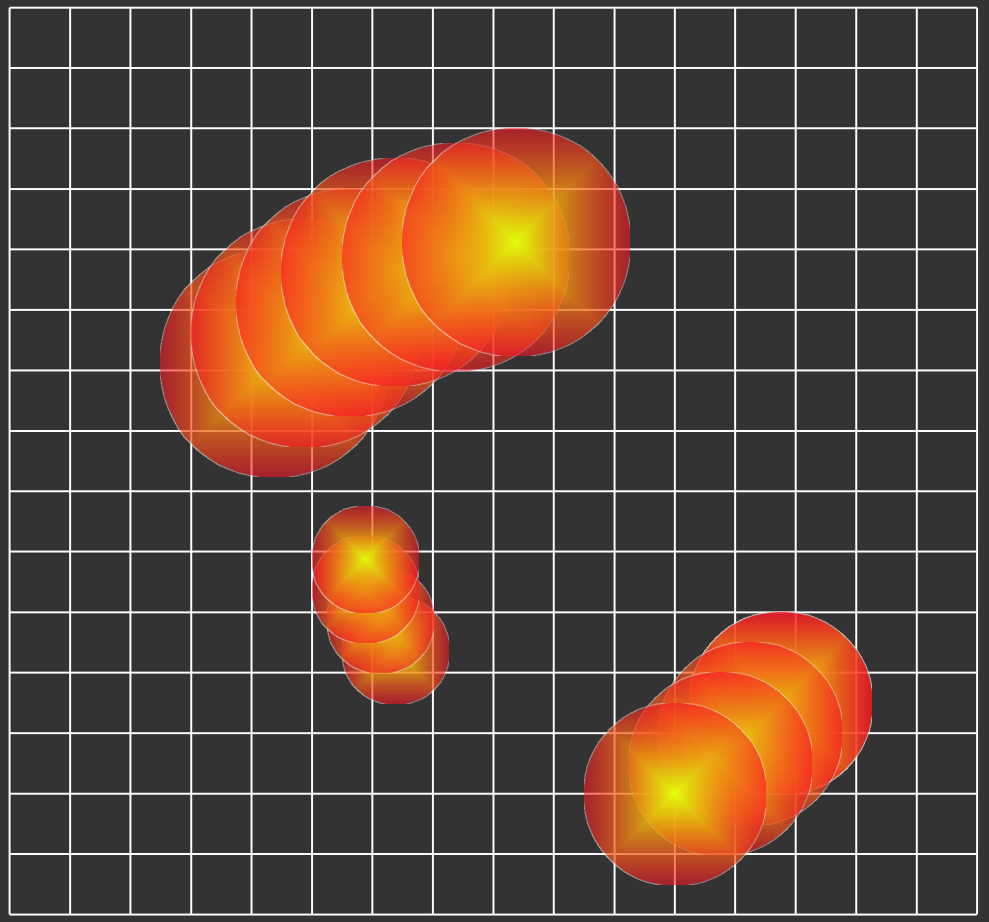
effects

image

Fourier transform

# convolutional gridding

$$V(u,v,w) = \int A(l,m,w)\, I(l,m)\, e^{-2\pi i[ul+vm]}\, dldm$$



# Working Model

IP[y]: Notebook imgmodel-nb Last Checkpoint: Dec 04 07:33 (unsaved changes)

File Edit View Insert Cell Kernel Help

Code Cell Toolbar: None

```
------------------------------------------------------------
Parameters
------------------------------------------------------------
Image size: 4608
PSF size: 50
W layers: 6
Compression: 0.447056121277
Visibilities (compressed) per output channel: 8.11139e+07 (2.596 GB)
```

### Compute resources

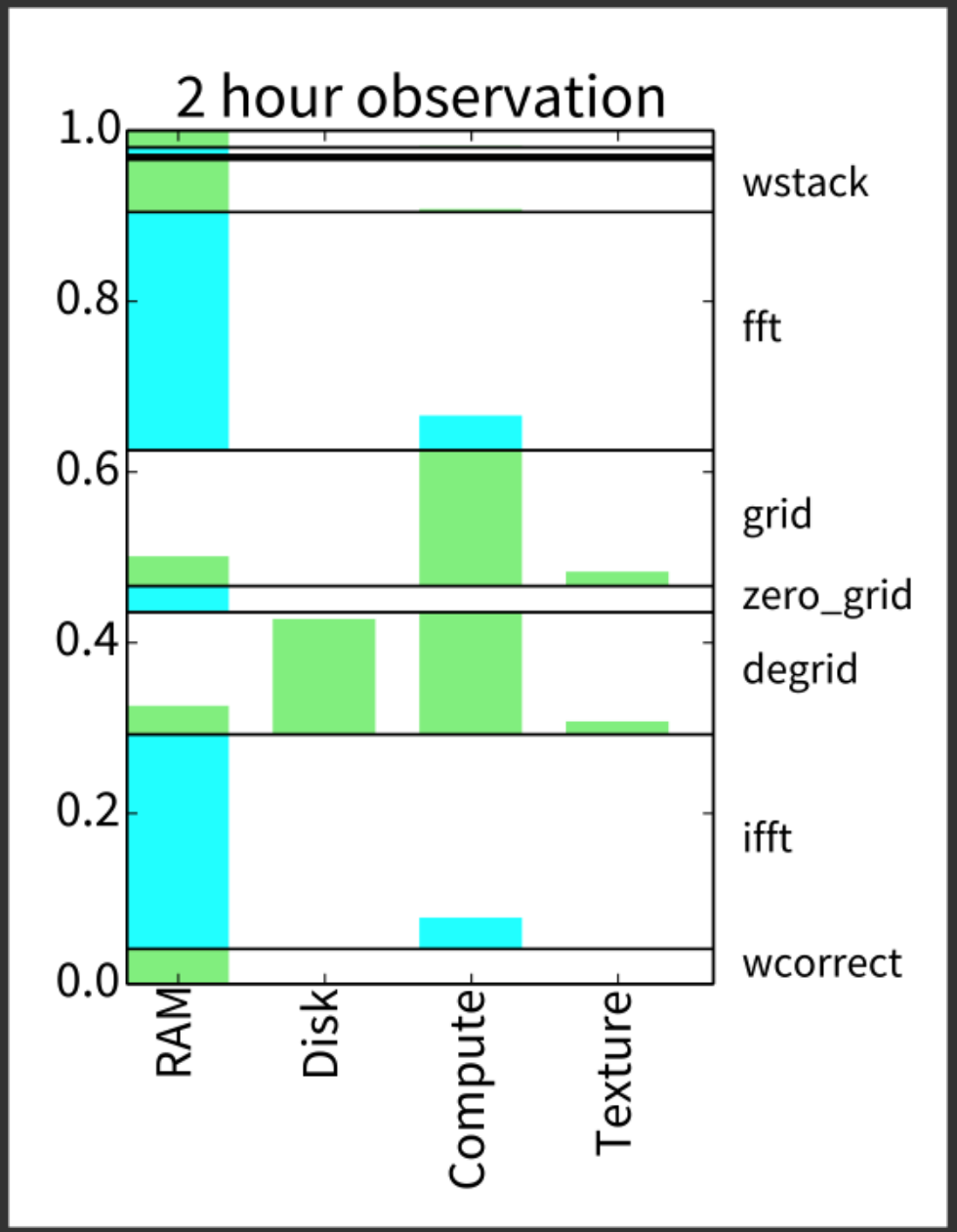
The number of GPUs required is worst when the observation time is minimum (2h). The results below are for a Tegra X1.

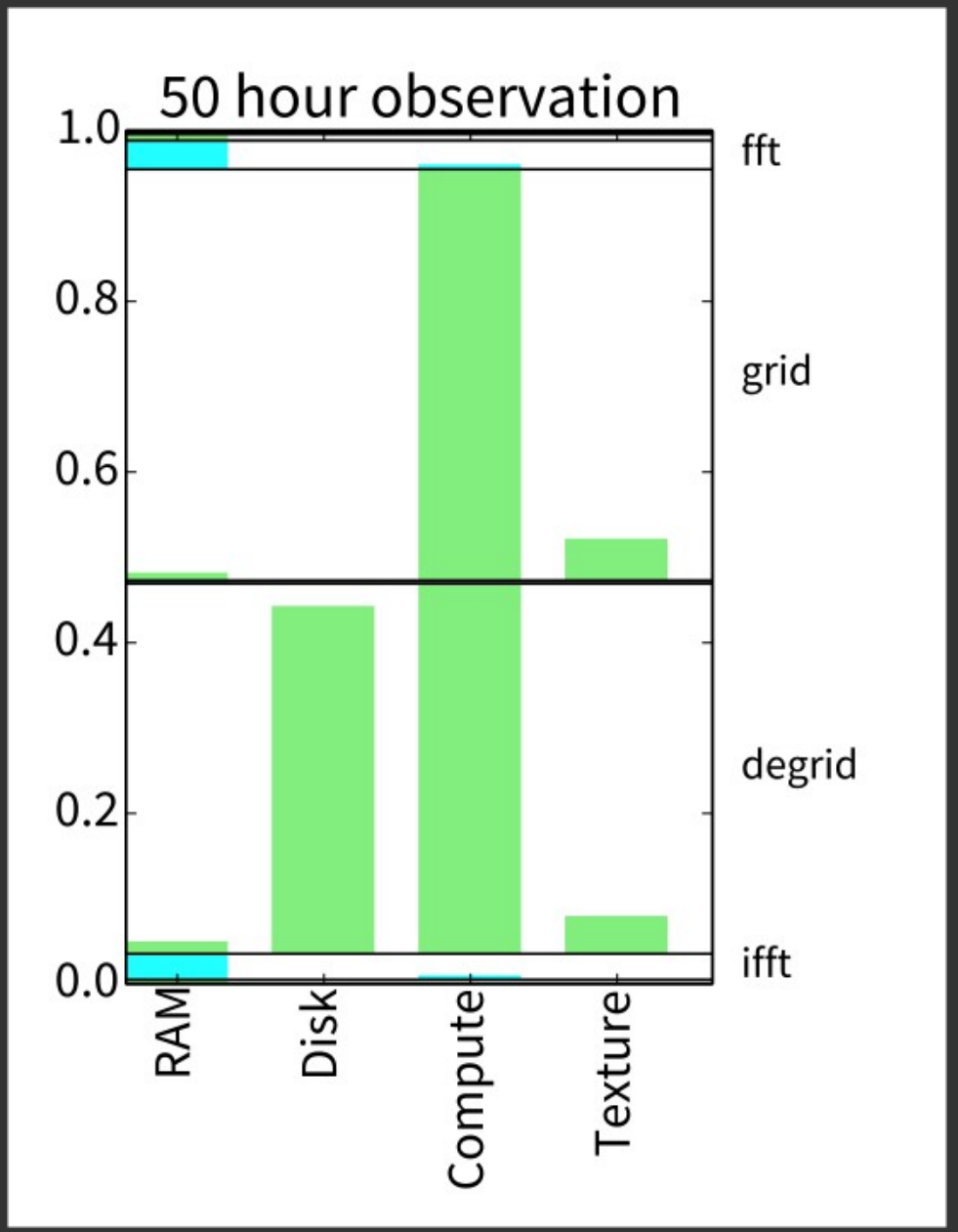
In [13]:
```
s_result = imgmodel.generate_models(s_args)
s_model = s_result.model
imgmodel.report_models('Timings (single output channel)', s_result, lambda x: x.report_times(device=s_device))
imgmodel.report_models('Buffers', s_result, Model.report_buffers)
imgmodel.report_models('Totals per output channel', s_result, lambda x: x.report_totals(device=s_device))
s_gpus = s_model.get_time(s_device) * s_args.out_channels / s_args.time
imgmodel.report_models('Grand totals', s_result,
    lambda x: x.report_totals(s_args.out_channels, 1.0, s_args.time))
print
print "GPUs required: {0:.1f}".format(s_gpus)
print
imgmodel.report_models('Totals per GPU', s_result,
    lambda x: x.report_totals(s_args.out_channels, s_gpus, device=s_device))
```

```
------------------------------------------------------------
Timings (single output channel)
------------------------------------------------------------
wcorrect: 2.687
ifft: 16.380
degrid: 9.344
zero_grid: 1.991
grid: 10.383
fft: 18.200
wstack: 3.981
initialise_peak_tiles: 0.166
find_peak: 0.203
subtract_psf: 0.586
update_peak: 1.280
```

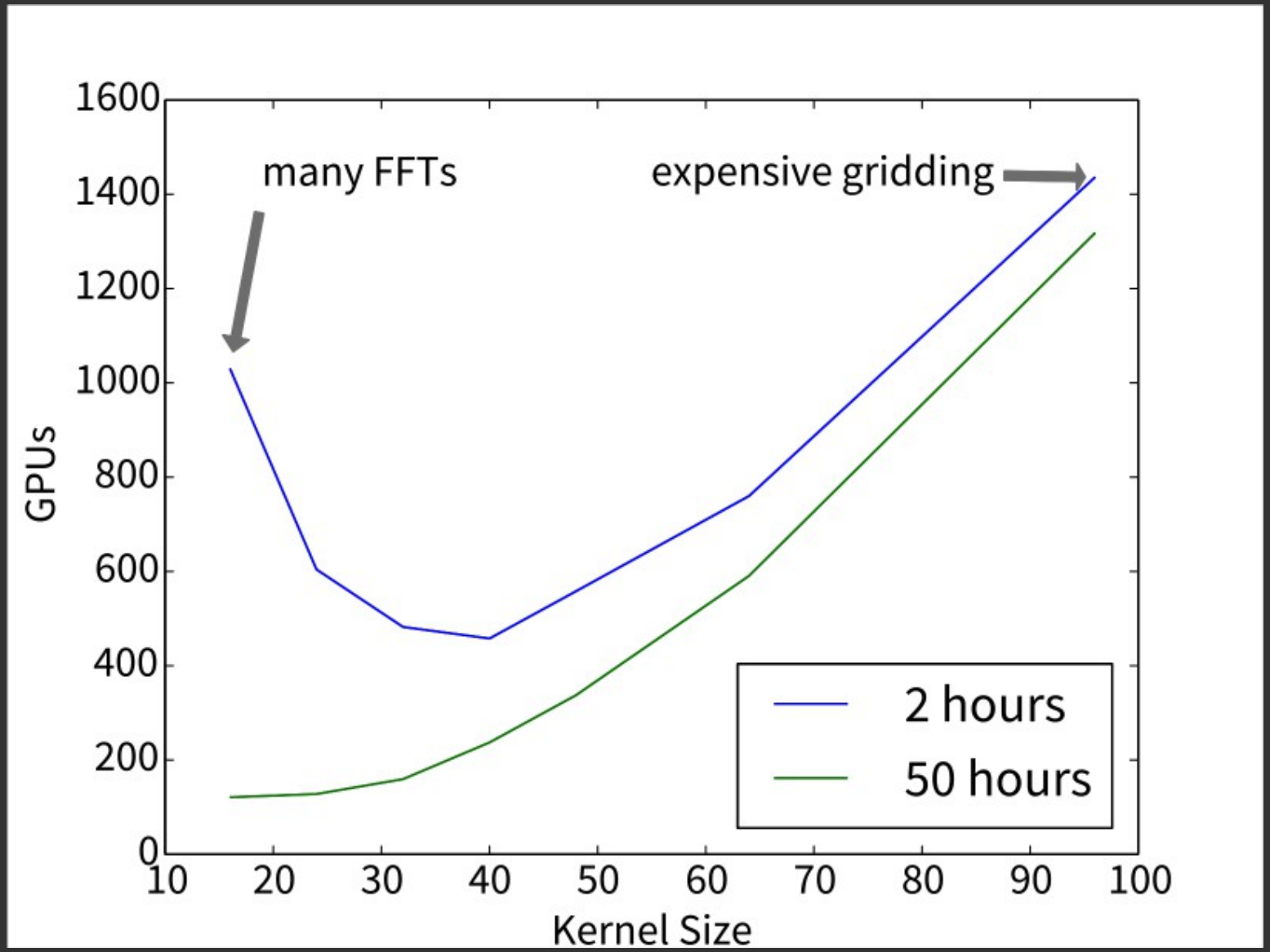
8 Hours, 64 Antennas, Single Channel, 4k Image

# I cannae push it any faster, Captain!

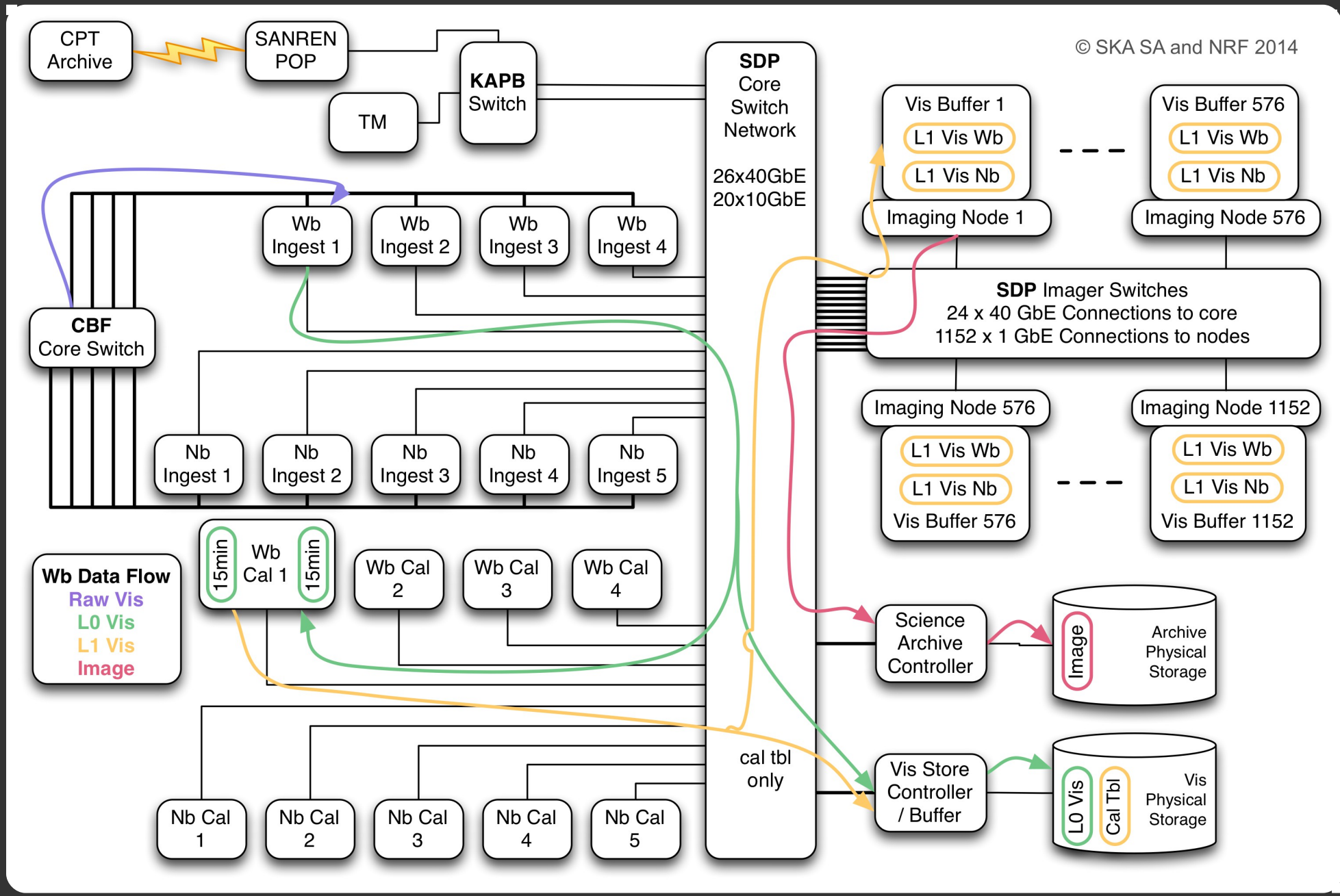


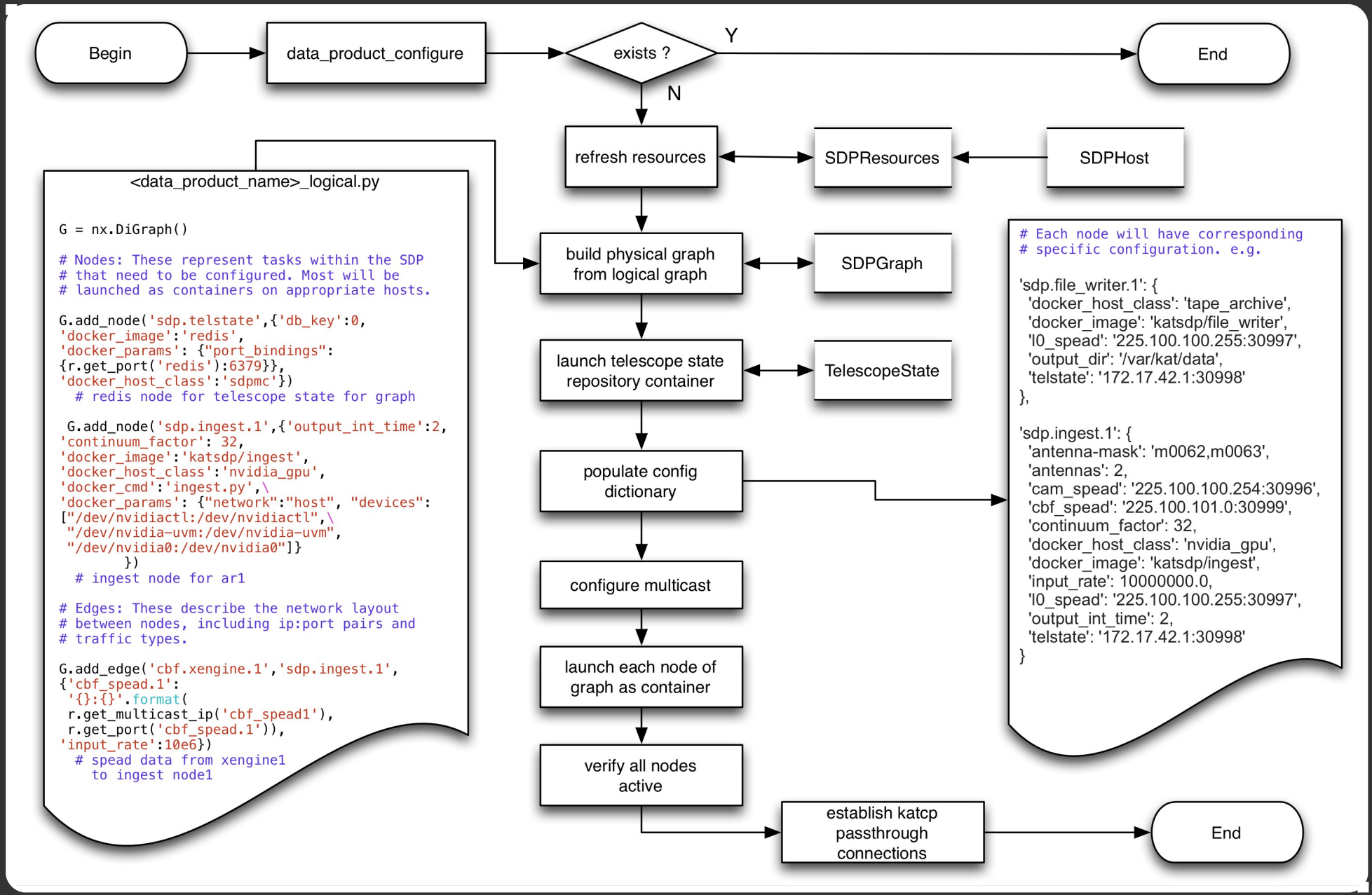
# Trade-off between gridding and FFT costs



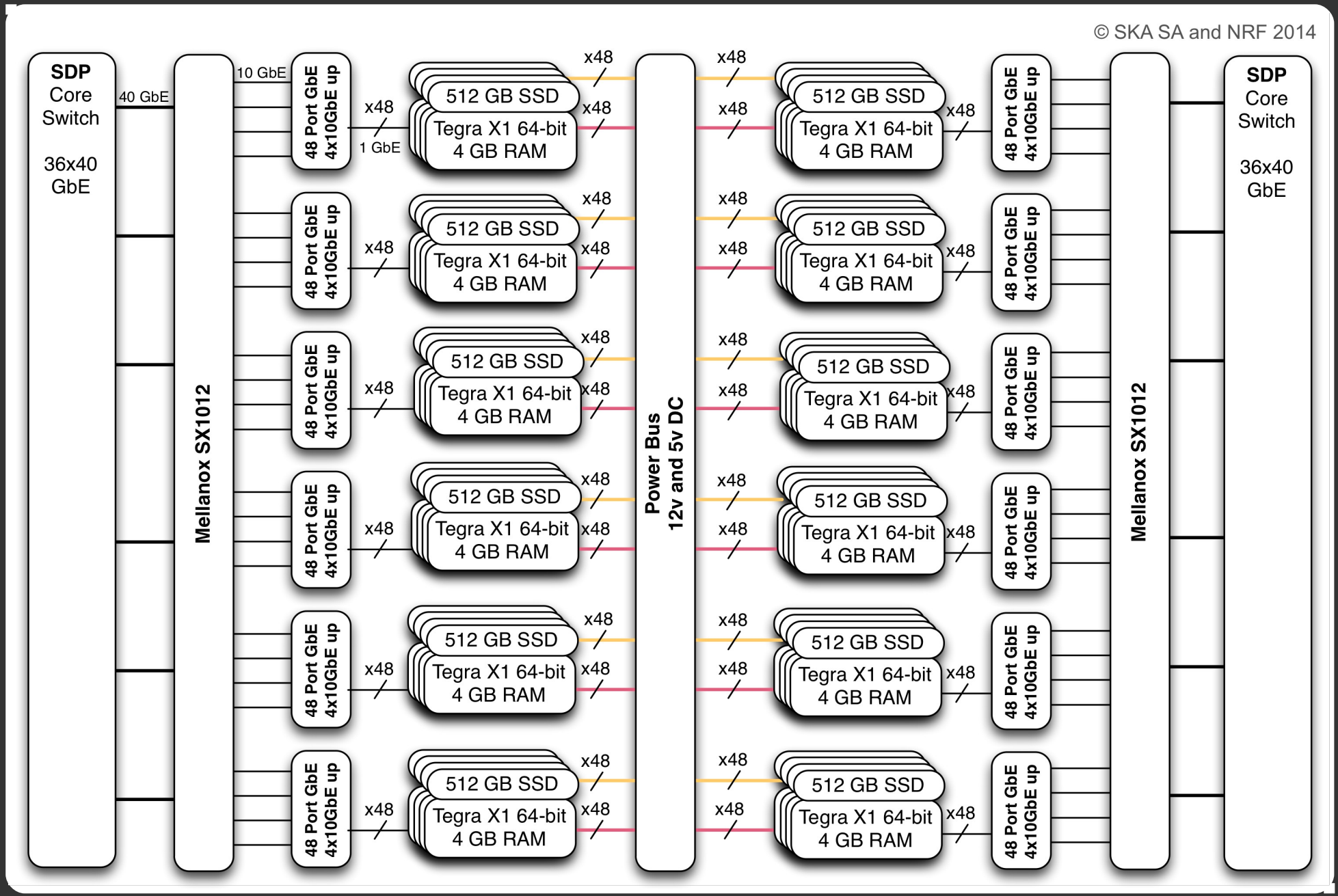
# Anatomy of a modern radio telescope



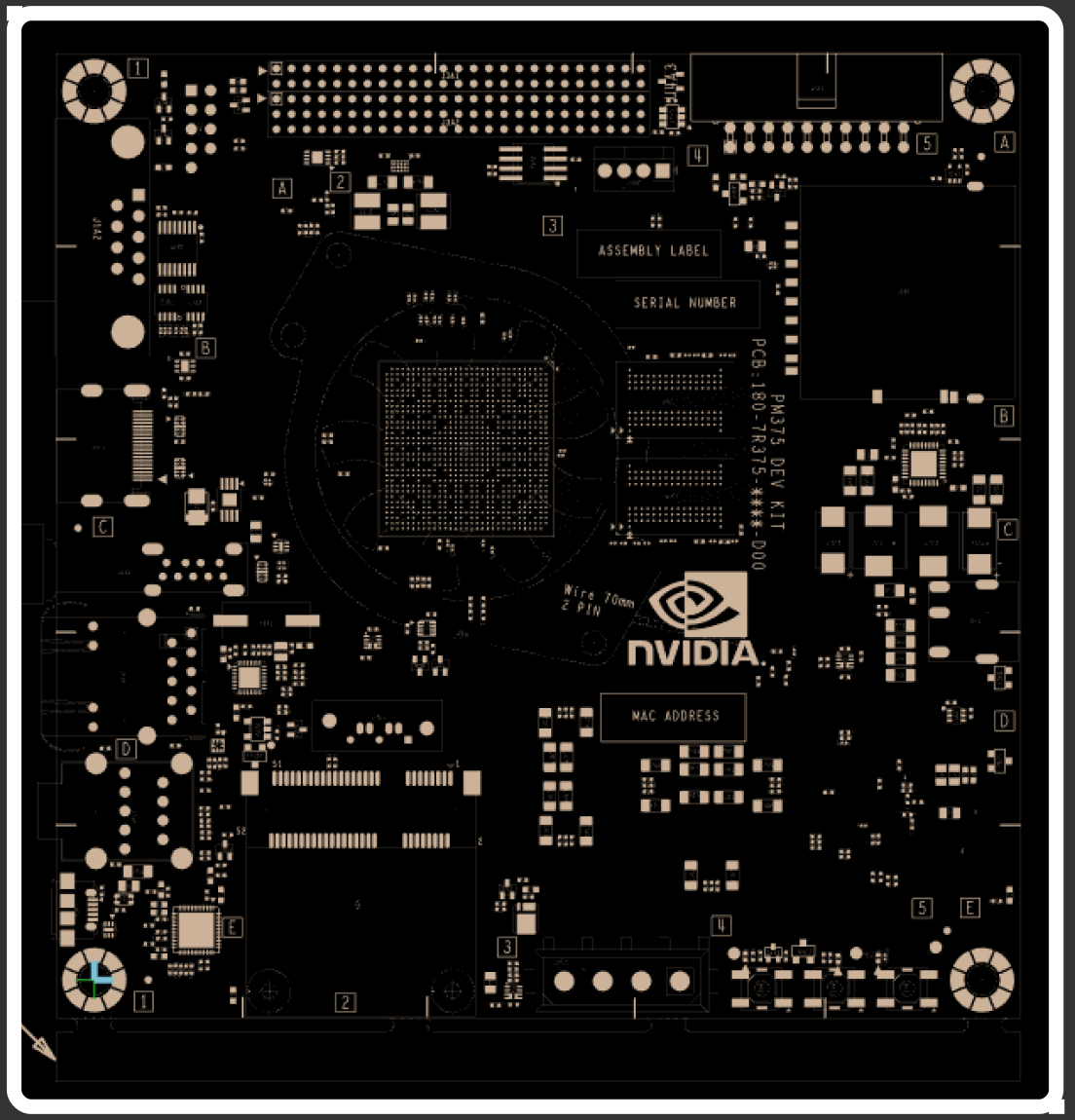
# Graphs all the way down until you hit the turtles...



# How embarrassing is your parallelism ?

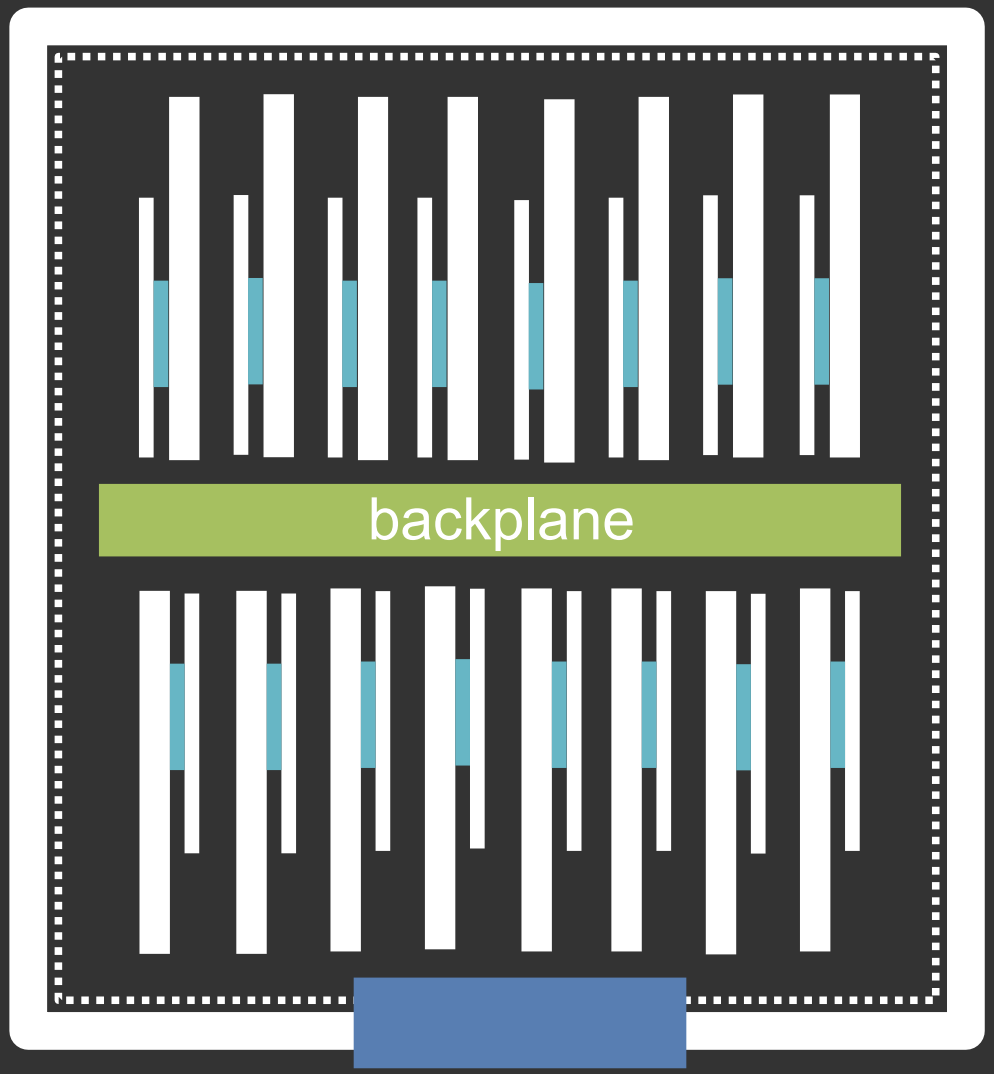


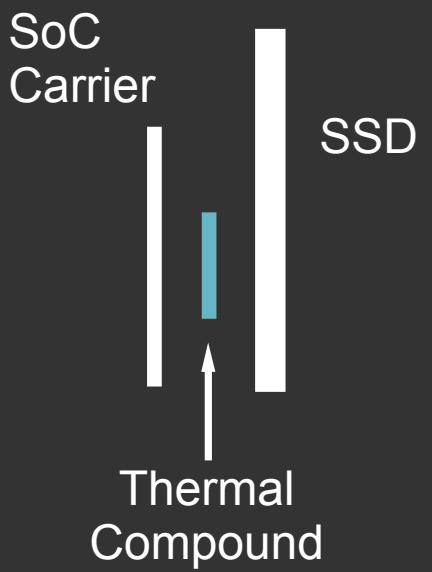
# ignore the logo, it's clearly an internal design.



# pre cambrian cooling inc.

100 Nodes
50 TFLOPS
0.4 TB RAM
50 TB SSD
40 Gbps Eth
0.7 kW

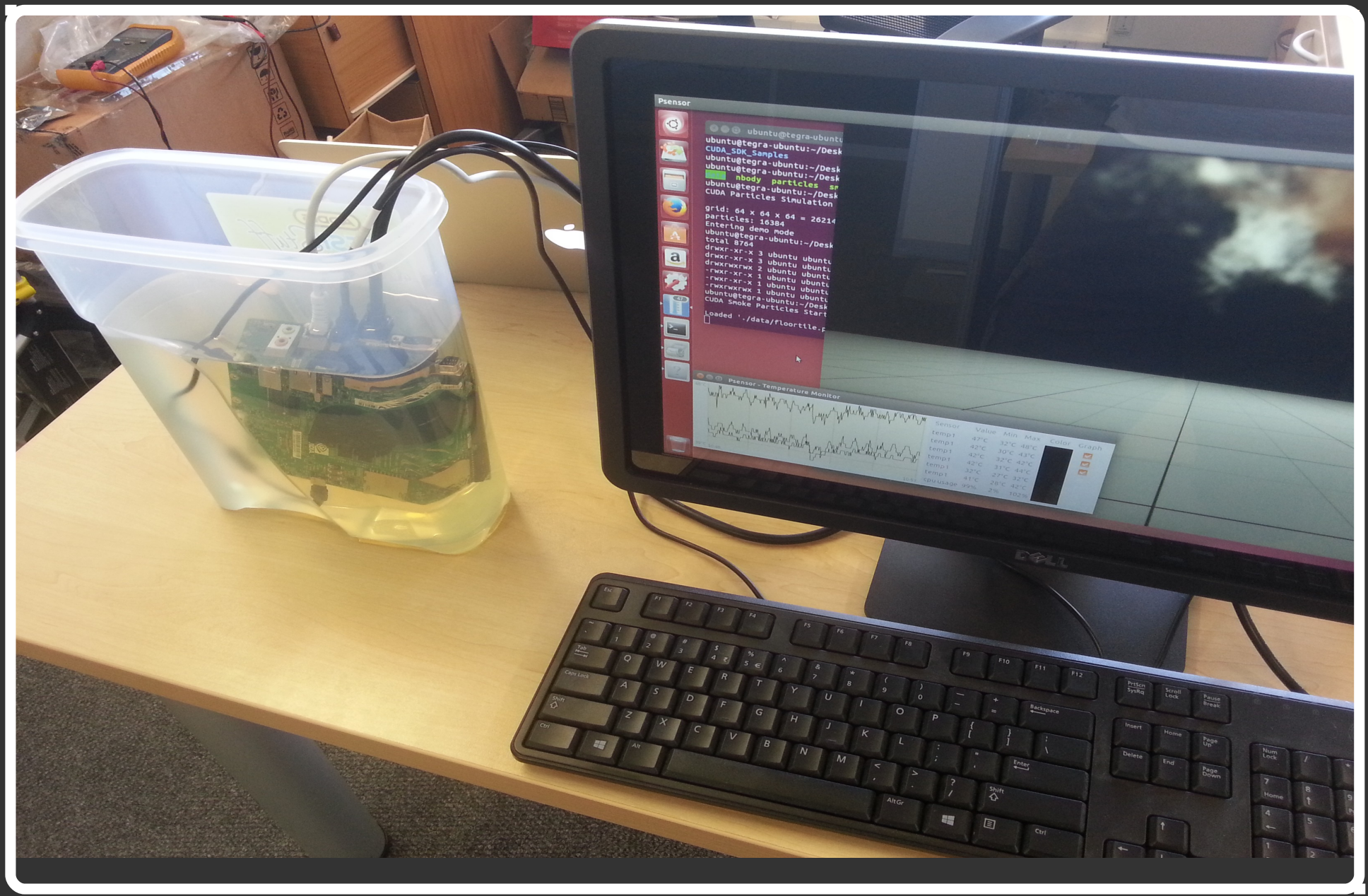



Hive
FERRO

Ground Loop

# deep fried


Psensor
ubuntu@tegra-ubuntu:~/Desk
CUDA_SDK_samples
ubuntu@tegra-ubuntu:~/Desk
ubuntu@tegra-ubuntu:~/Desk
nbody particles
ubuntu@tegra-ubuntu:~/Desk
CUDA Particles Simulation
grid: 64 x 64 x 64 = 26214
particles: 16384
Entering demo mode
ubuntu@tegra-ubuntu:~/Desk
total 8764
drwxr-xr-x 3 ubuntu ubuntu
drwxr-xr-x 3 ubuntu ubuntu
drwxrwxrwx 2 ubuntu ubuntu
-rwxr-xr-x 1 ubuntu ubuntu
-rwxr-xr-x 1 ubuntu ubuntu
-rwxrwxrwx 1 ubuntu ubuntu
ubuntu@tegra-ubuntu:~/Desk
CUDA Smoke Particles Start
Loaded './data/floortile.p
Psensor - Temperature Monitor
DELL

# The build

## TEGRA X1

**1056 Nodes**

Tegra X1
4 GB RAM
512 GB SSD

**22 Switches**

2 x 10 GbE SFP+
48 x 1 GbE

**11 Pods**

15M Ground Loop
50L Mineral Oil

## TESLA K40

**50 Servers**

2 x Tesla K40
2 x E5-2660v3
6 x 2TB SATA
64 GB RAM

**3 Switches**

4 x 40 GbE QSFP
36 x 10 GbE SFP+

**3 Racks**

Just a rack

# Super green ? Super green.

## TEGRA X1

**$350** kilo

$310k Hardware
$40k Infrastructure

**12.4** kW

11.9 kW Hardware
0.5 kW Cooling

## TESLA K40

**$1,056** kilo

$816k Hardware
$58k Infrastructure

**57.5** kW

44.3 kW Hardware
13.2 kW Cooling