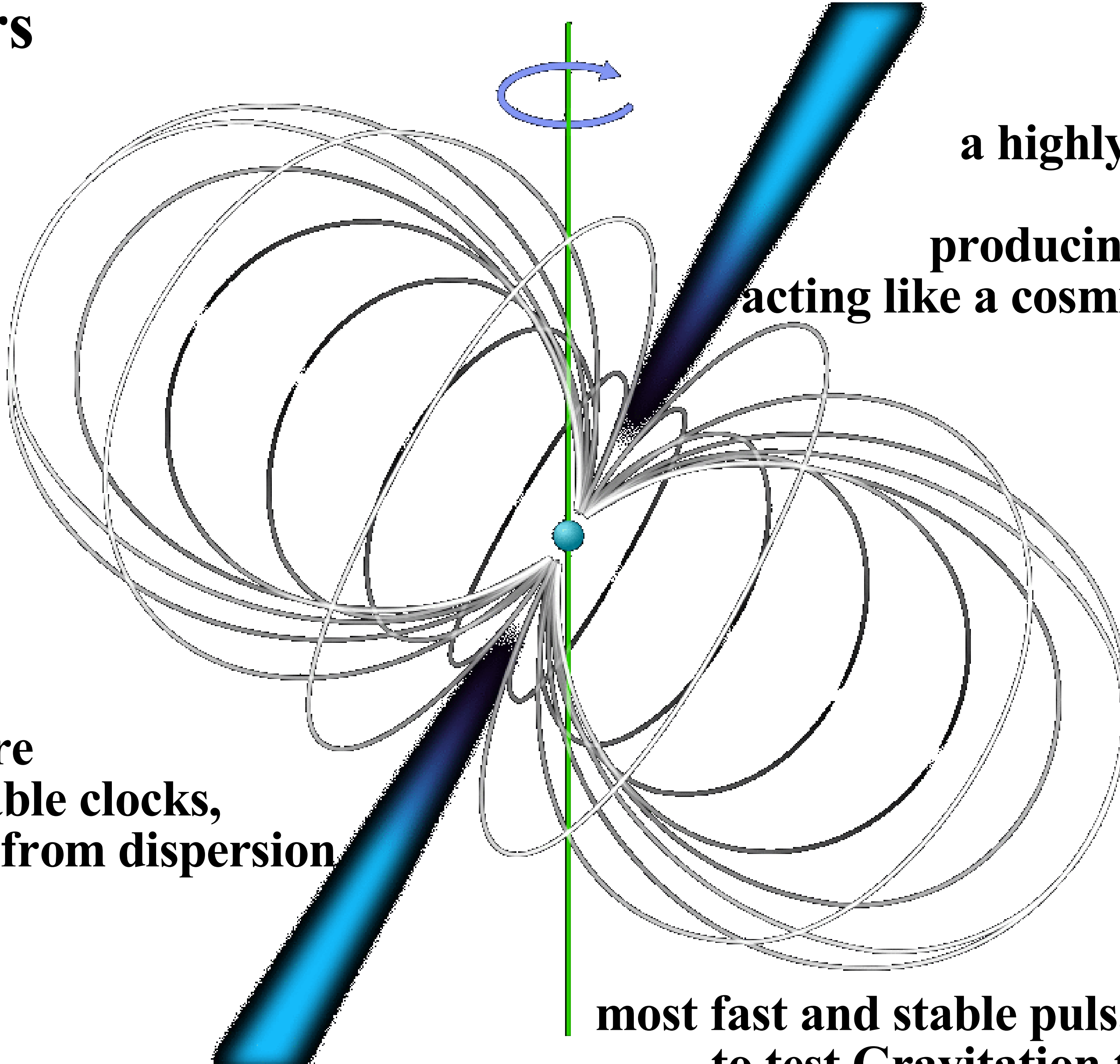
Real-Time Pulsar Timing Signal Processing on GPUs

Plan :

- **Pulsar Timing**
- **Instrumentations**

LPC2E, CNRS Orléans - FRANCE -
Ismaël Cognard,
Gilles Theureau,
Grégory Desvignes,
Dalal Ait-Allal

Pulsars

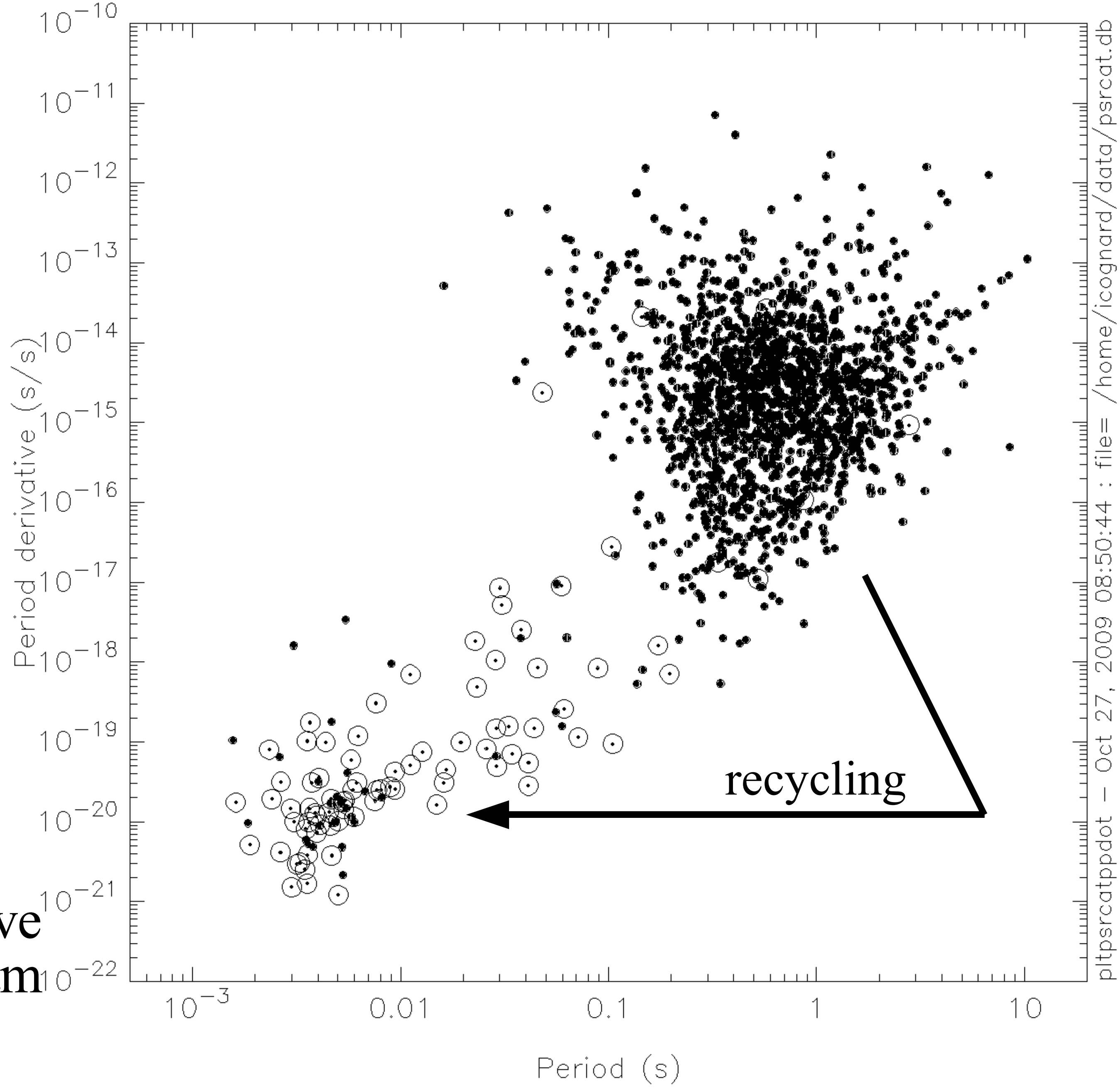


a highly magnetized
neutron star
producing two beams
acting like a cosmic lighthouse

pulsars are
highly stable clocks,
suffering from dispersion

most fast and stable pulsars are used
to test Gravitation theories and
to probe the Cosmological Gravitational Wave background

Pulsars



Period - Period derivative diagram

Five stable pulsars timed at Nançay

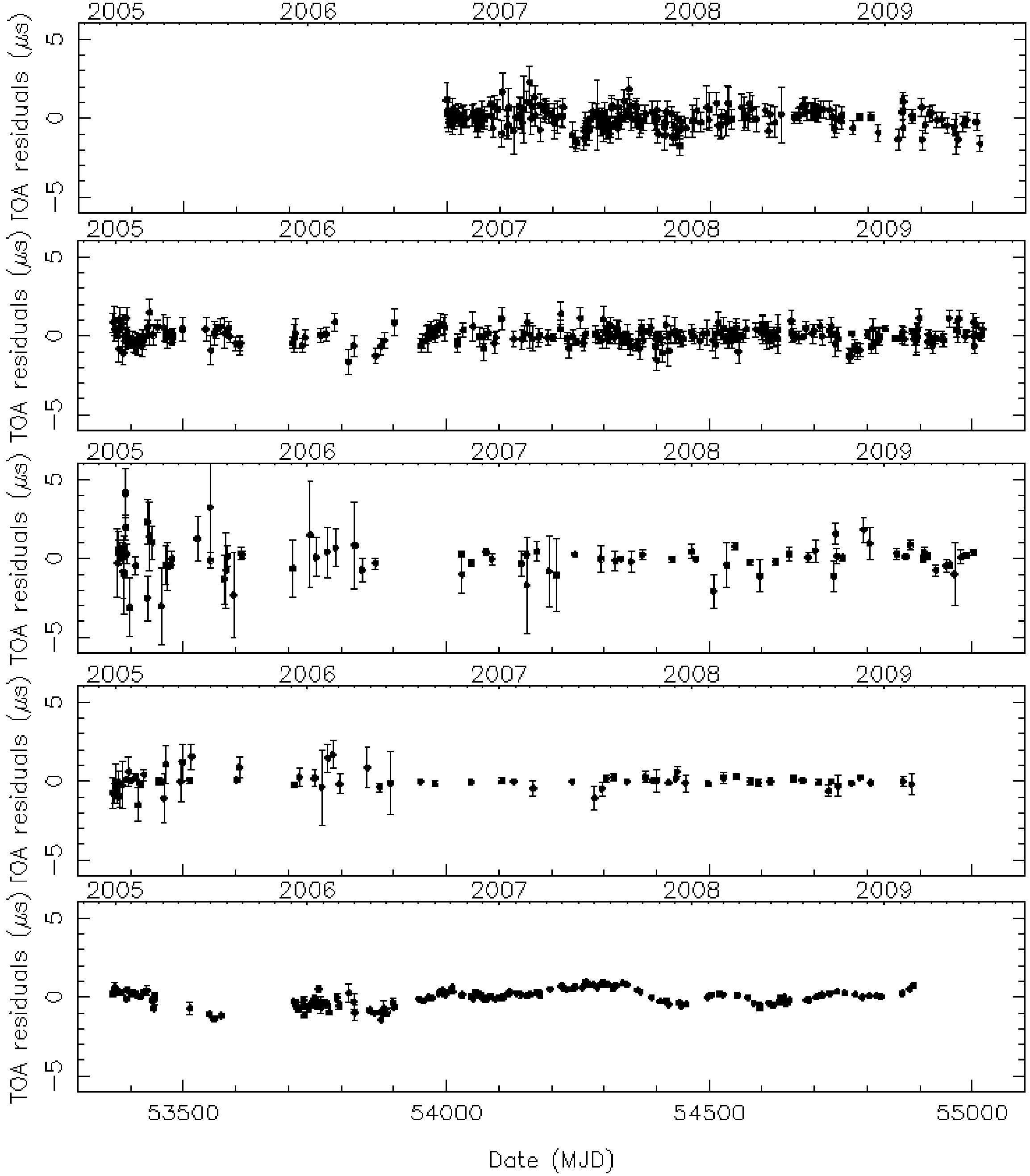
PSR J1600-3053
rms 576ns

PSR J1713+0747
rms 350ns

PSR J1744-1134
rms 343ns

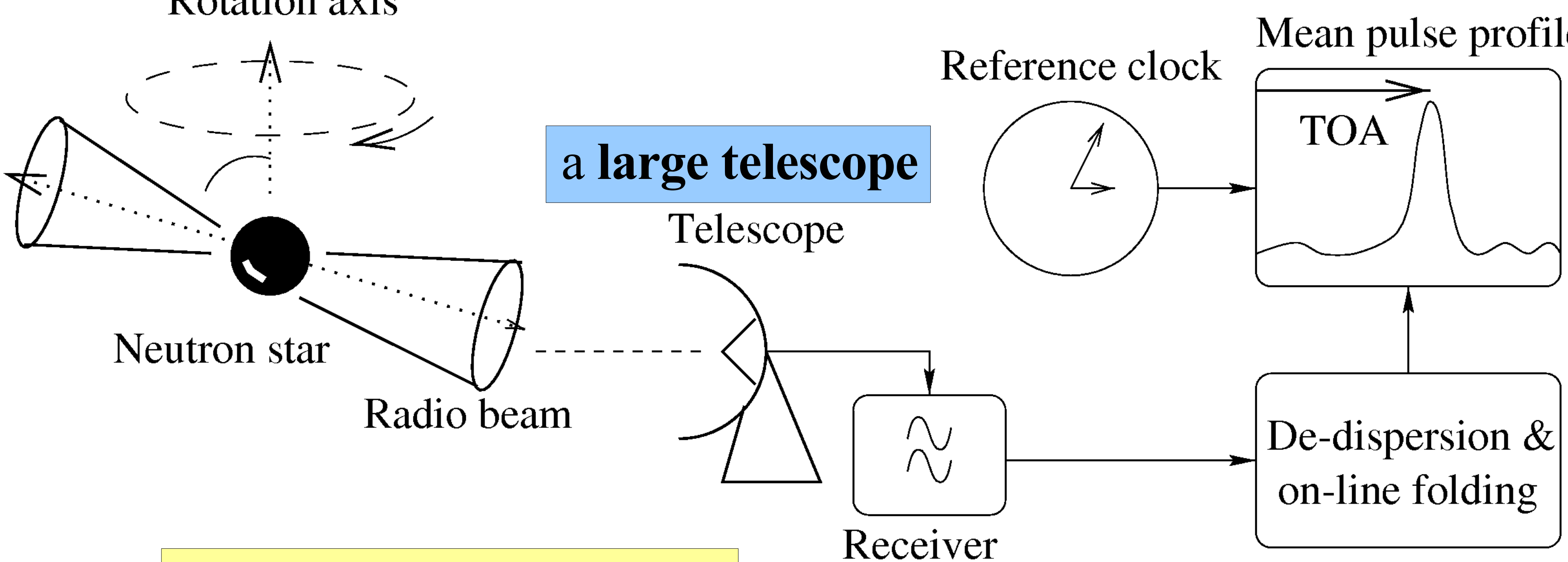
PSR J1909-3744
rms 109ns

PSR B1937+21
rms 387ns



Pulsar timing

(Lorimer & Kramer, 2004)

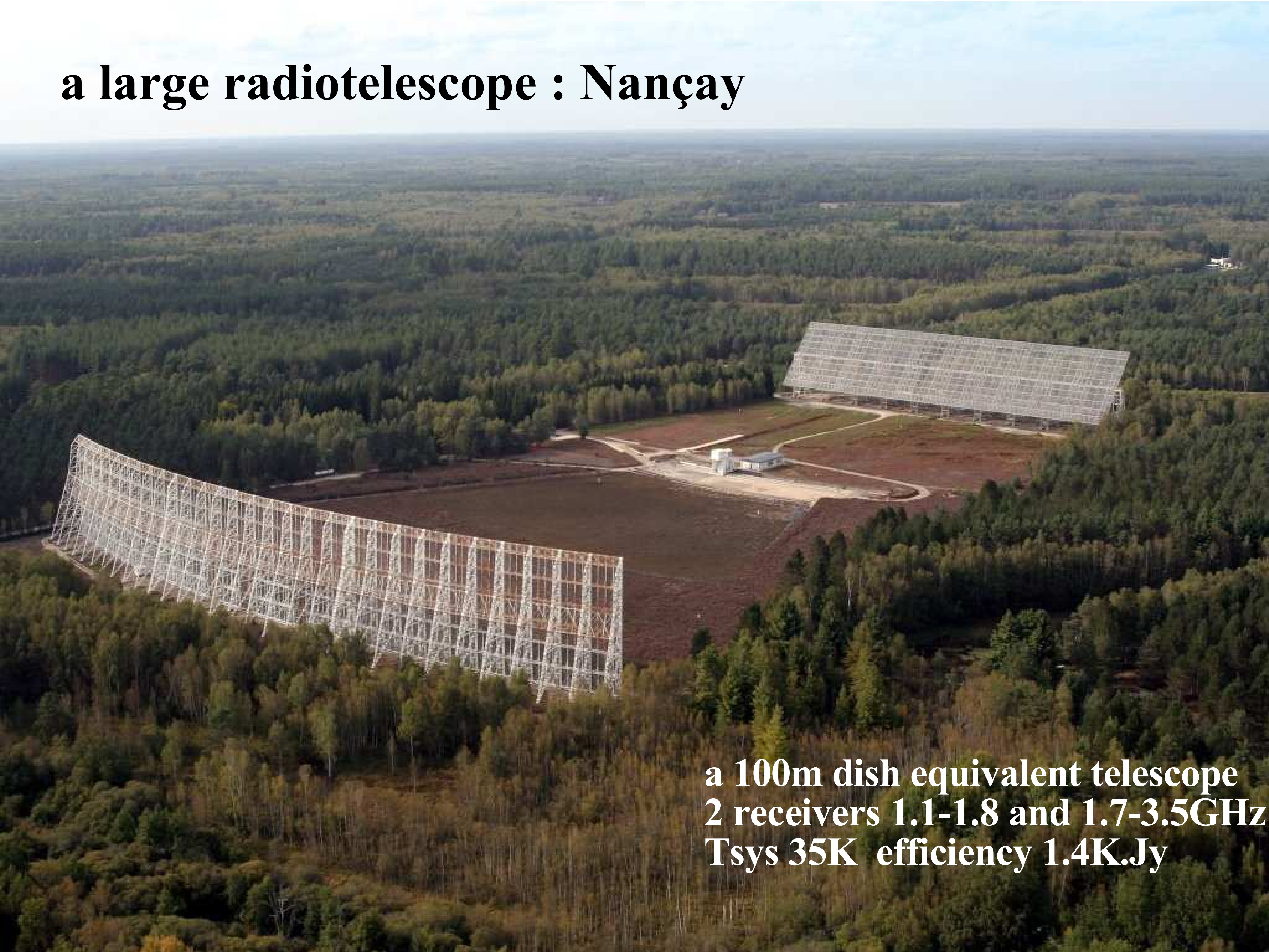


a **large telescope**

the coherent dedispersion removes the **dispersion effect** of the interstellar medium directly on recorded voltages

a **huge computing power** is required to coherently dedisperse over a large bandwith

a large radiotelescope : Nançay



a 100m dish equivalent telescope
2 receivers 1.1-1.8 and 1.7-3.5GHz
Tsys 35K efficiency 1.4K.Jy

dedispersion
a bit of history...

1988 swept LO (co 8MHz)
~ 0.5µs uncertainty on PSR B1937+21

1996 filterbank NBPP (inco 120MHz)
Navy Berkeley Pulsar Processor

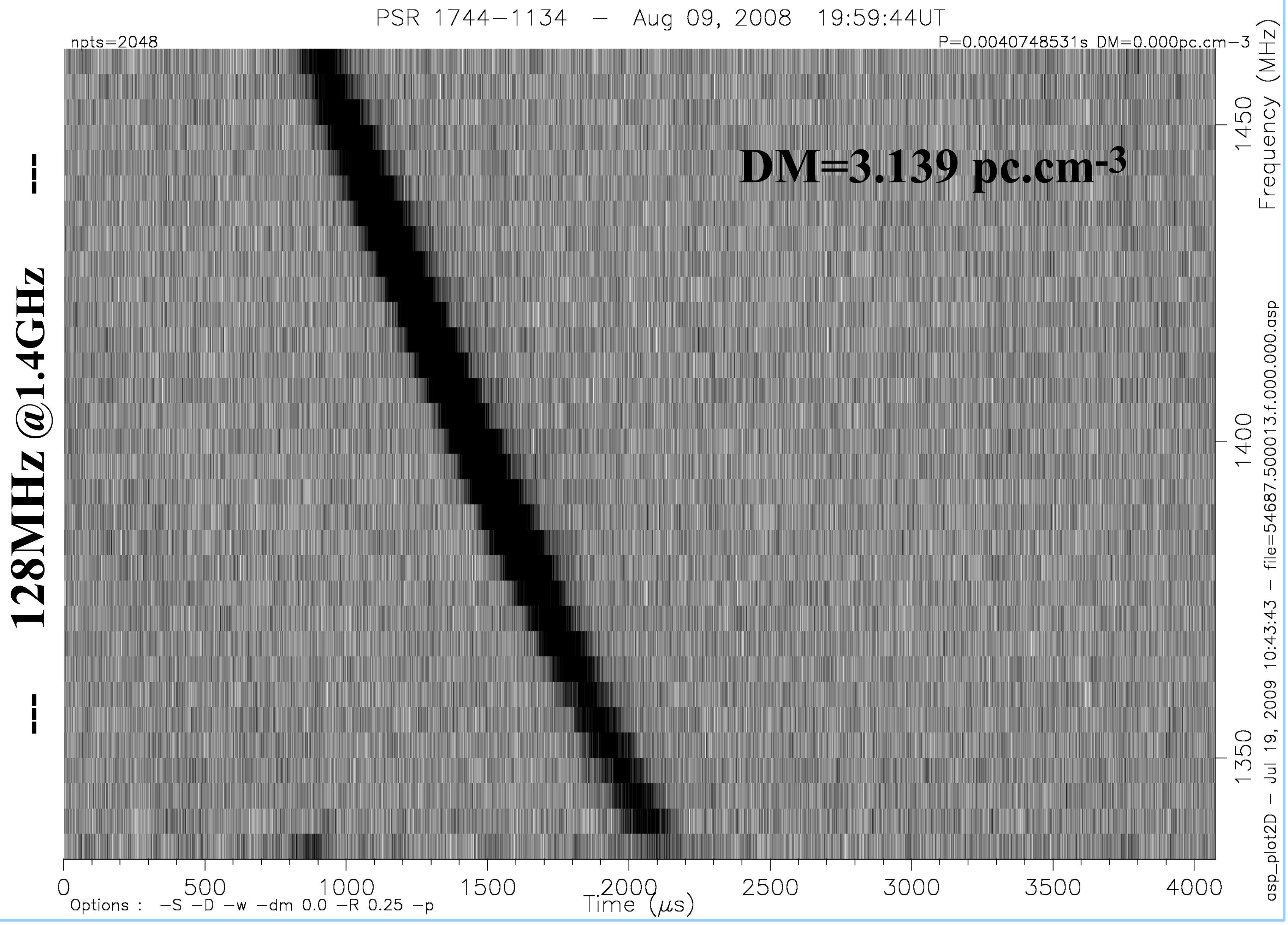
2004 coherent dedispersor BON (co 64-128MHz)
Berkeley Orleans Nancay

2008 GPU coherent BON (co 128MHz)

swept LO

NBPP

Dispersion by the ISM



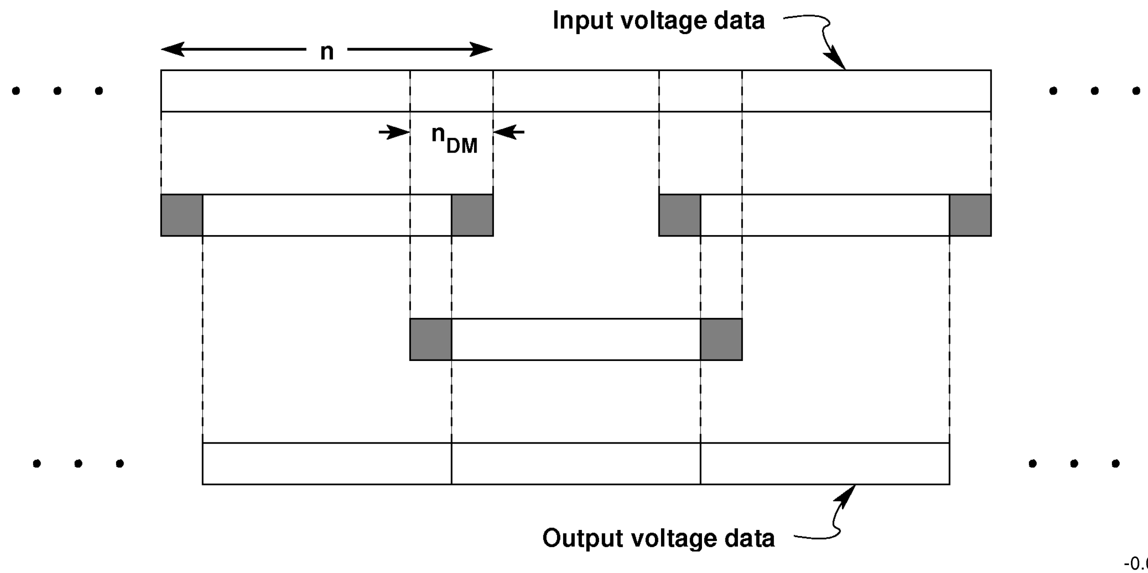
Coherent dedispersion

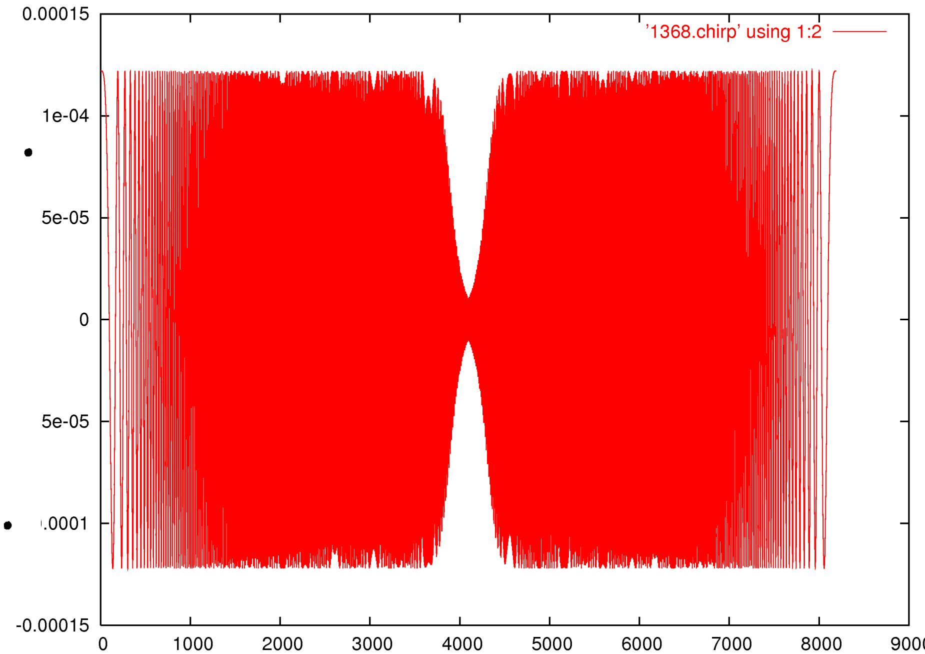
dispersion by the ionized ISM is a phase-only filter

coherent dedispersion

apply an inverse transfer function in the frequency domain

FFT + multiply + FFT^{-1} (+overlap)



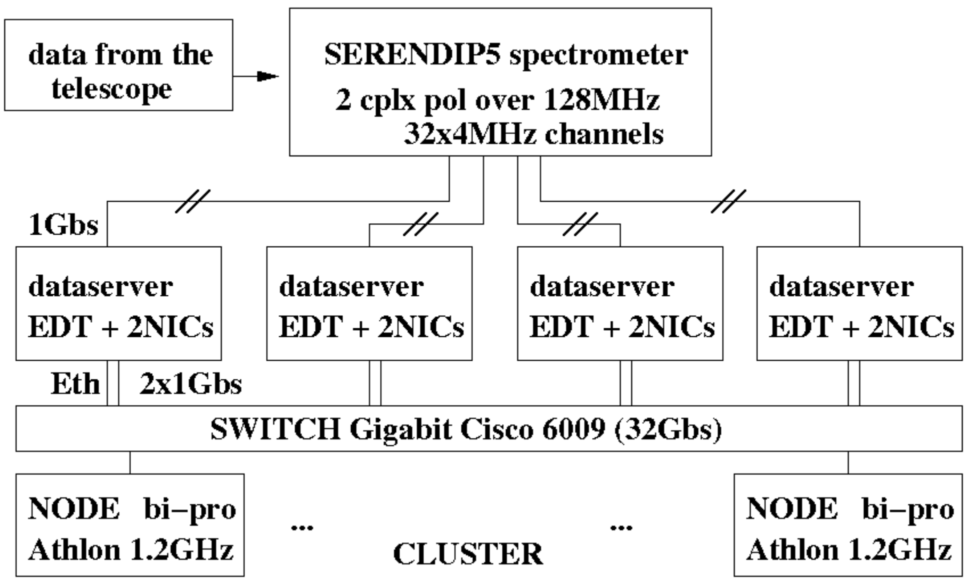


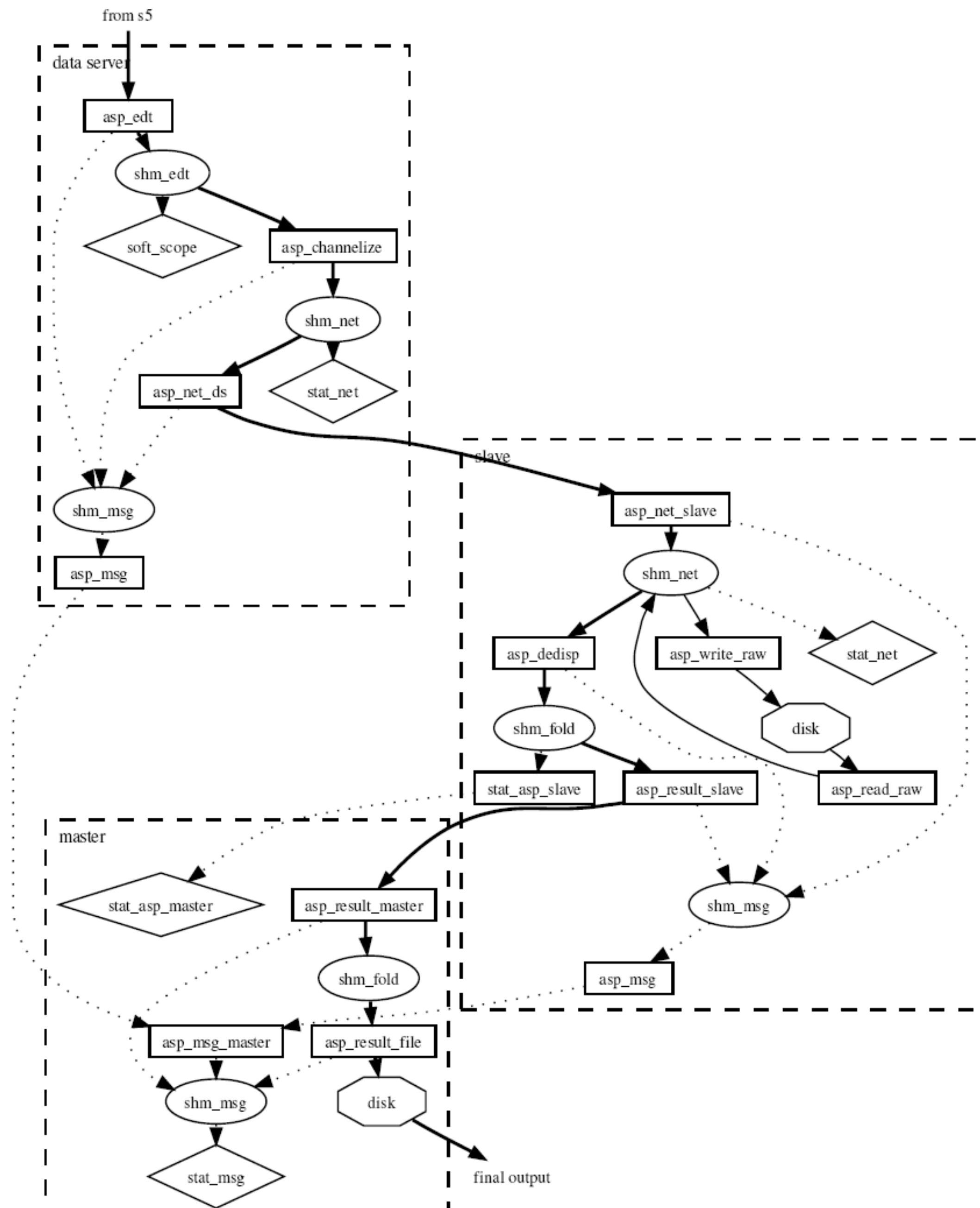
BON : a coherent dedispersion with a cluster of CPUs

64 bi-Athlon 1.2GHz
Gigabit fiber network
installed in 2001

Serendip5

BON : architecture

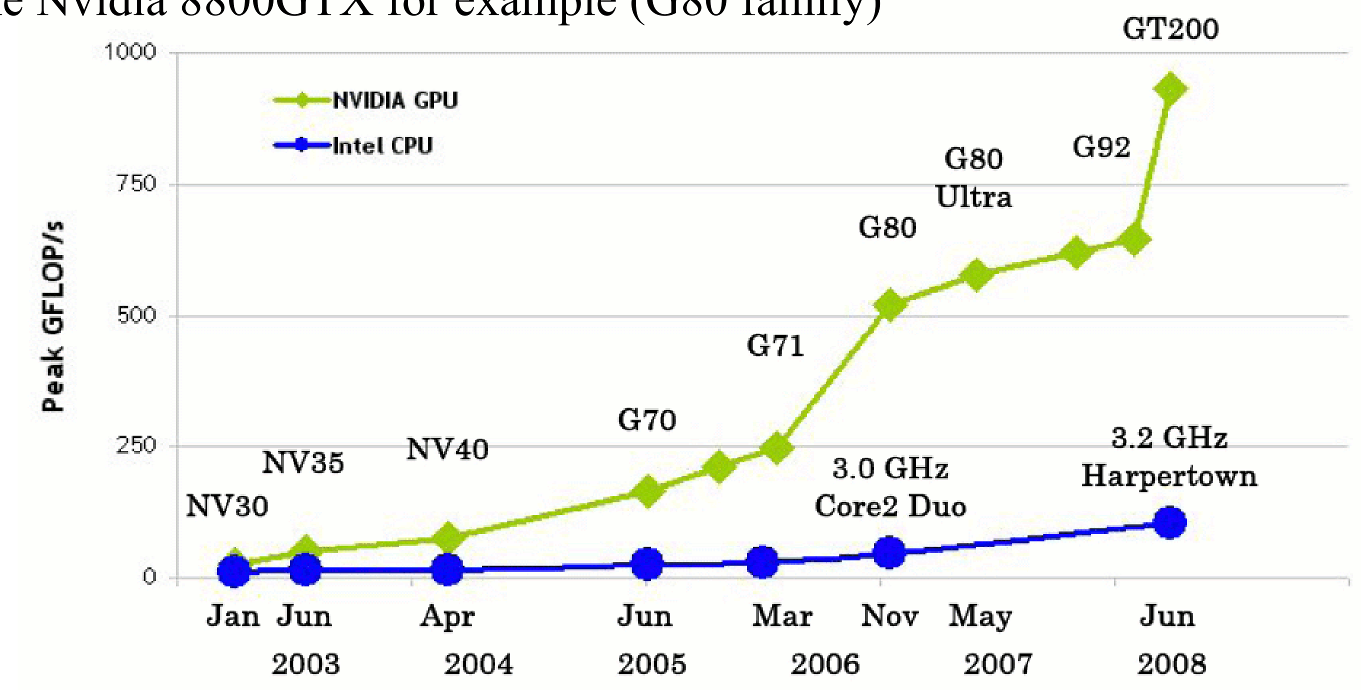




Upgrade of the BON dedispersor

in 2008,
it was time to think about the **upgrade** of the 7 years old cluster
and of the 4 years old instrumentation...

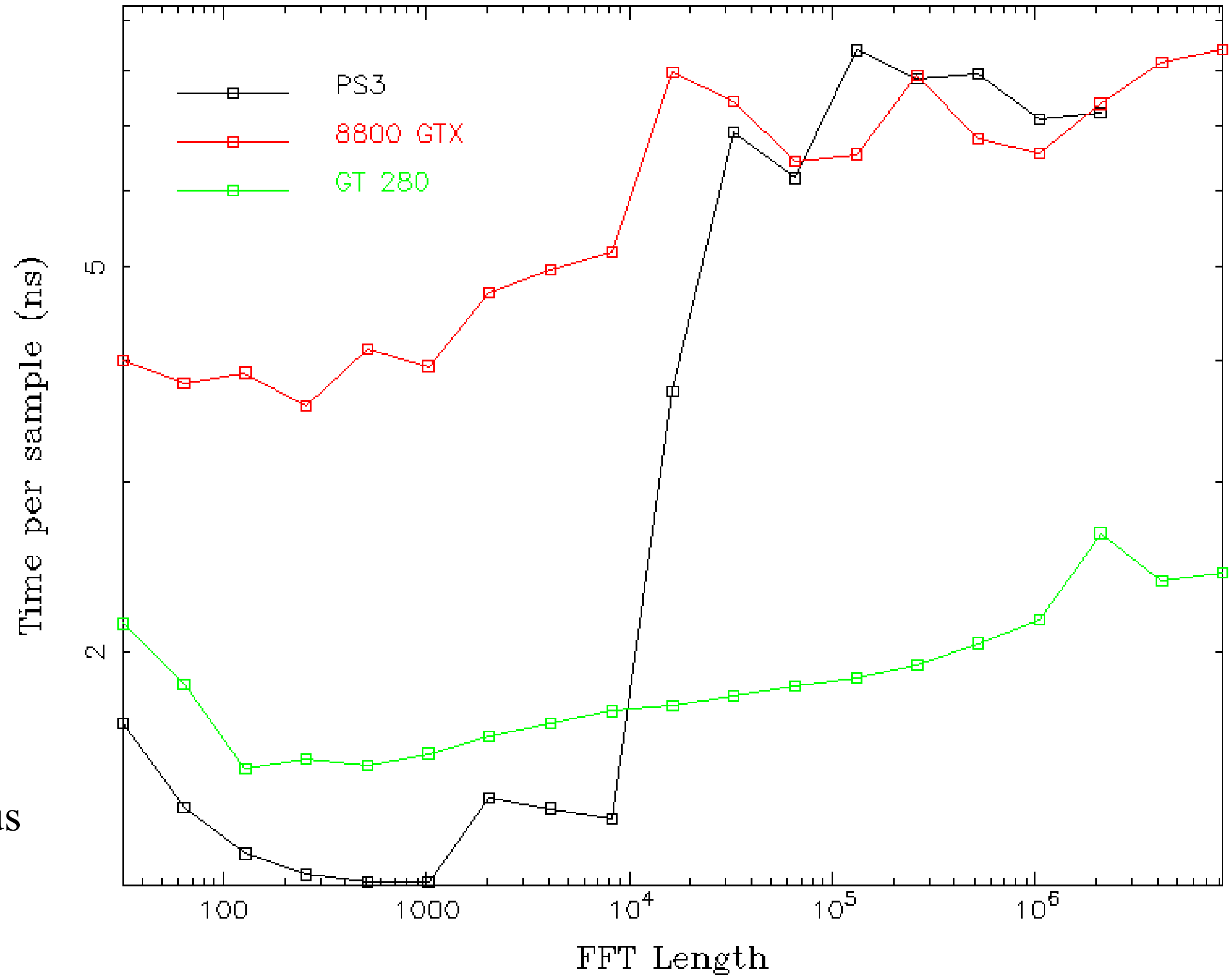
following preliminary tests by Paul Demorest, UC Berkeley, in 2007,
we tried different types of processors in addition to standard CPUs :
the GPUs with the Nvidia 8800GTX for example (G80 family)



GT200 = GeForce GTX 280	G71 = GeForce 7900 GTX	NV35 = GeForce FX 5950 Ultra
G92 = GeForce 9800 GTX	G70 = GeForce 7800 GTX	NV30 = GeForce FX 5800
G80 = GeForce 8800 GTX	NV40 = GeForce 6800 Ultra	

FFT tests of different engines

so, we tested two types of processors in addition to standard CPUs :
the IBM Cell (PlayStation3) and the Nvidia 8800GTX



PS3 is really good
but very limited by
the only Gigabit link it has
(15-20MHz bandwidth)

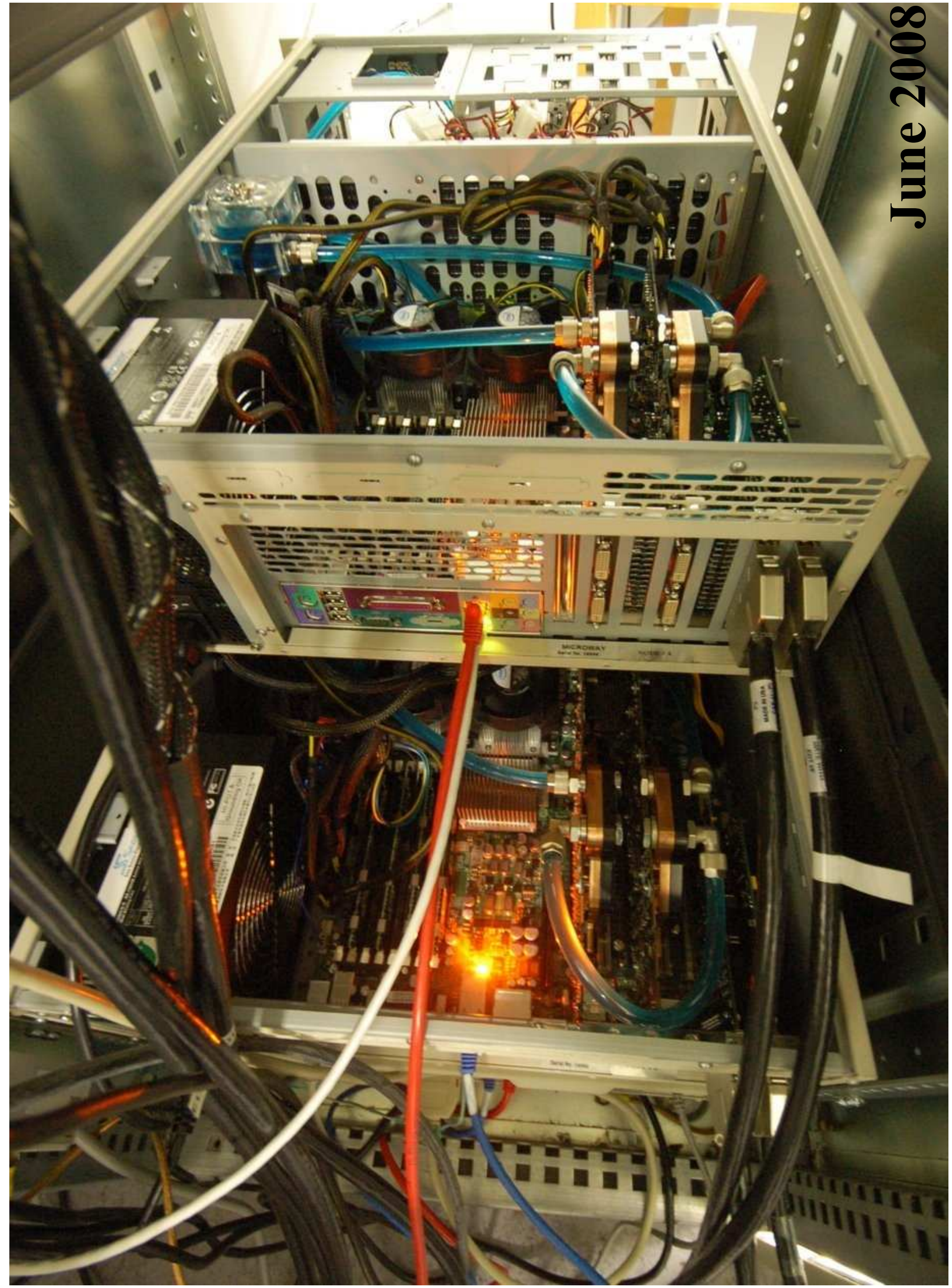
Nvidia 8800GTX is fairly good
and less limited by the PCIe bus
(few Gbs easily)

a GPU-based pulsar instrumentation
developped at Nançay

1 motherboard
    Supermicro X7DWA-N
2 CPUs
    quad-core Xeon
        E5420 (2.5GHz, 12MB cache)
4 GB main memory
2 PCI-CDa DMA interface (for EDT)
2 GeForce 8800GTX
and water cooling

+ CUDA librairies + CUFFT

this 5k€ system is able to
remove the **dispersion effect**
of the interstellar medium
directly on recorded voltages
over a 128MHz bandwidth
(the data rate is 2x 2Gbs)



June 2008

Code (excerpt)

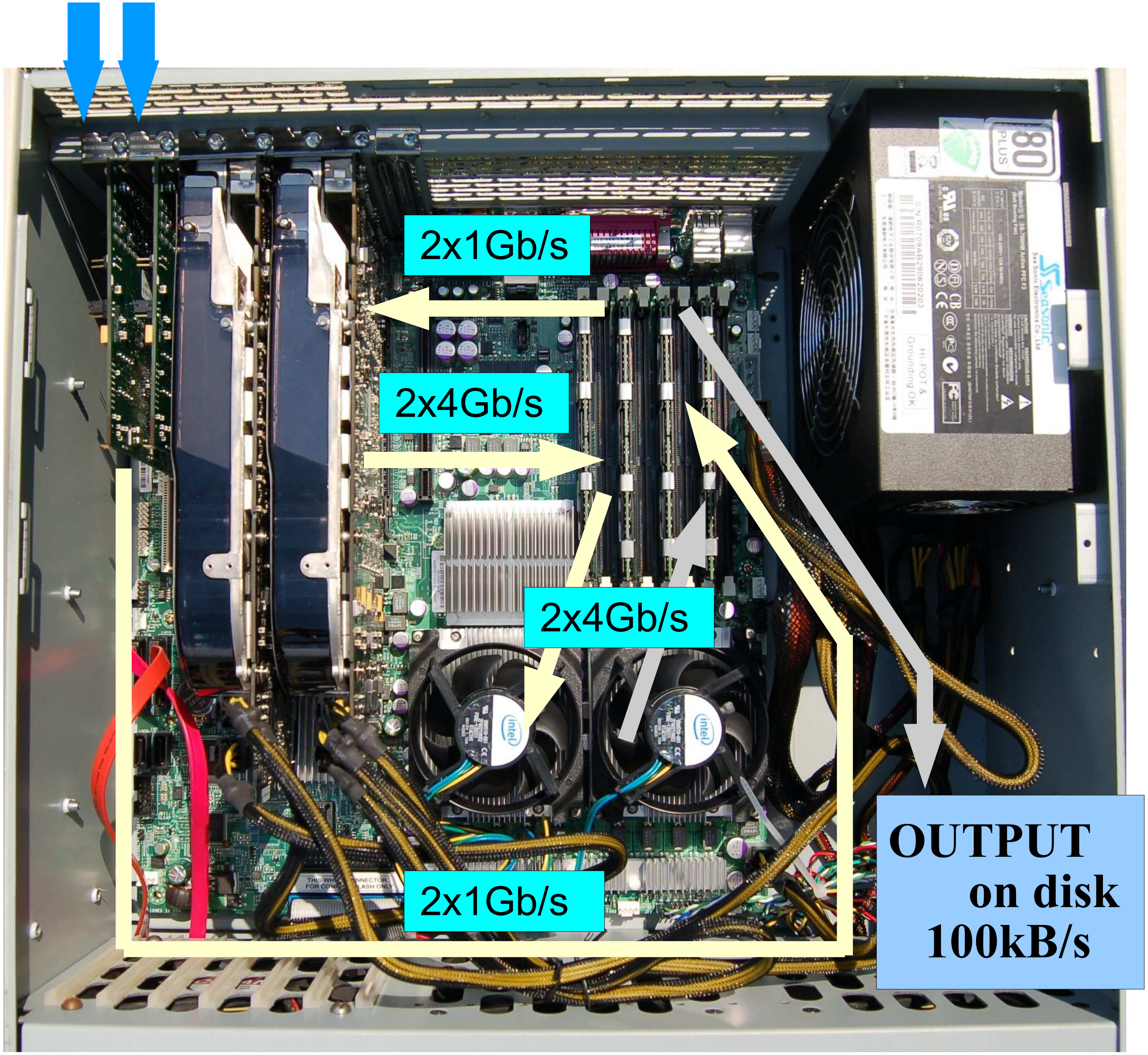
```
// -- Allocate and transfer of chirp function --
cudaMalloc((void**)&chirp_device, mem_chirp);
cudaMemcpy(chirp_device, chirp, mem_chirp, cudaMemcpyHostToDevice);

// -- CUFFT plan --
CUFFT_SAFE_CALL(cufftPlan1d(&plan,obs_params.fft_len,CUFFT_C2C,2*NFFTinSend));

// -- Transfer data --
cudaMemcpyAsync(tbuf_device, tbuf_host, mem_tbuf, cudaMemcpyHostToDevice,0);
CUDA_SAFE_CALL(cudaThreadSynchronize());

// -- Forward FFT --
CUFFT_SAFE_CALL(cufftExecC2C(plan,fftbuf_device,fftbuf_device,CUFFT_FORWARD));

// -- Chirp filter multiply --
vec_mult_complex<<<nb_mult,512>>>
    ( (float2 *)&fftbuf_device[i*obs_params.fft_len],
    (float2 *)&chirp_device[obs_params.fft_len*freq_chan],
    obs_params.fft_len, 2*NFFTinSend );

// -- Inverse FFT --
CUFFT_SAFE_CALL(cufftExecC2C(plan,fftbuf_device,fftbuf_device,CUFFT_INVERSE));

// -- Detect and Stokes --
detect_4pol<<<32,64>>>((float2 *)fftbuf_device,
    (float2 *)fftbuf_mid, obs_params.fft_len*NFFTinSend);
```

Code (detect_4pol kernel)

```
/* Device function to compute squares, cross-products in-place
 * Inputs:
 *   vx = (X_I, X_Q, ... ) n times
 *   vy = (Y_I, Y_Q, ... ) n times
 *   n = number of data points
 * Outputs:
 *   vx = (XX, YY, ... ) n times
 *   vy = (Re(X*Y), Im(X*Y), ... ) n times
 */
__global__ void detect_4pol(float2 *vx, float2 *vy, size_t n) {
    const int nt = blockDim.x * gridDim.x;
    const int tid = blockIdx.x * blockDim.x + threadIdx.x;
    float2 pp, xp;
    for (int i=tid; i<n; i+=nt) {
        pp.x = vx[i].x*vx[i].x + vx[i].y*vx[i].y; // XX
        pp.y = vy[i].x*vy[i].x + vy[i].y*vy[i].y; // YY
        xp.x = vx[i].x*vy[i].x + vx[i].y*vy[i].y; // Re(X*Y)
        xp.y = vx[i].x*vy[i].y - vx[i].y*vy[i].x; // Im(X*Y)
        vx[i] = pp;
        vy[i] = xp;
    }
}
```

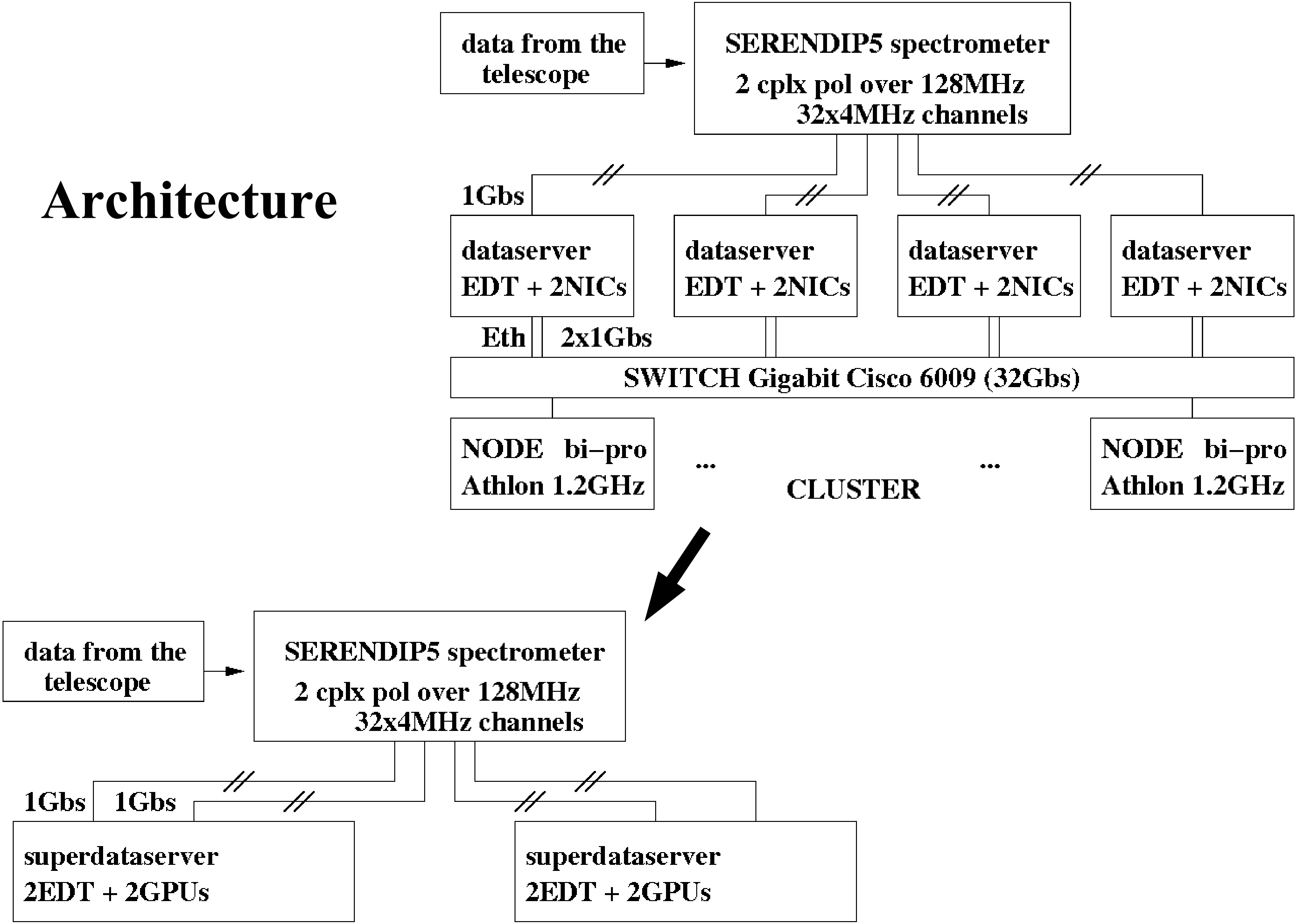
INPUT at 2Gb/s from Serendip5 spectrometer

the data path

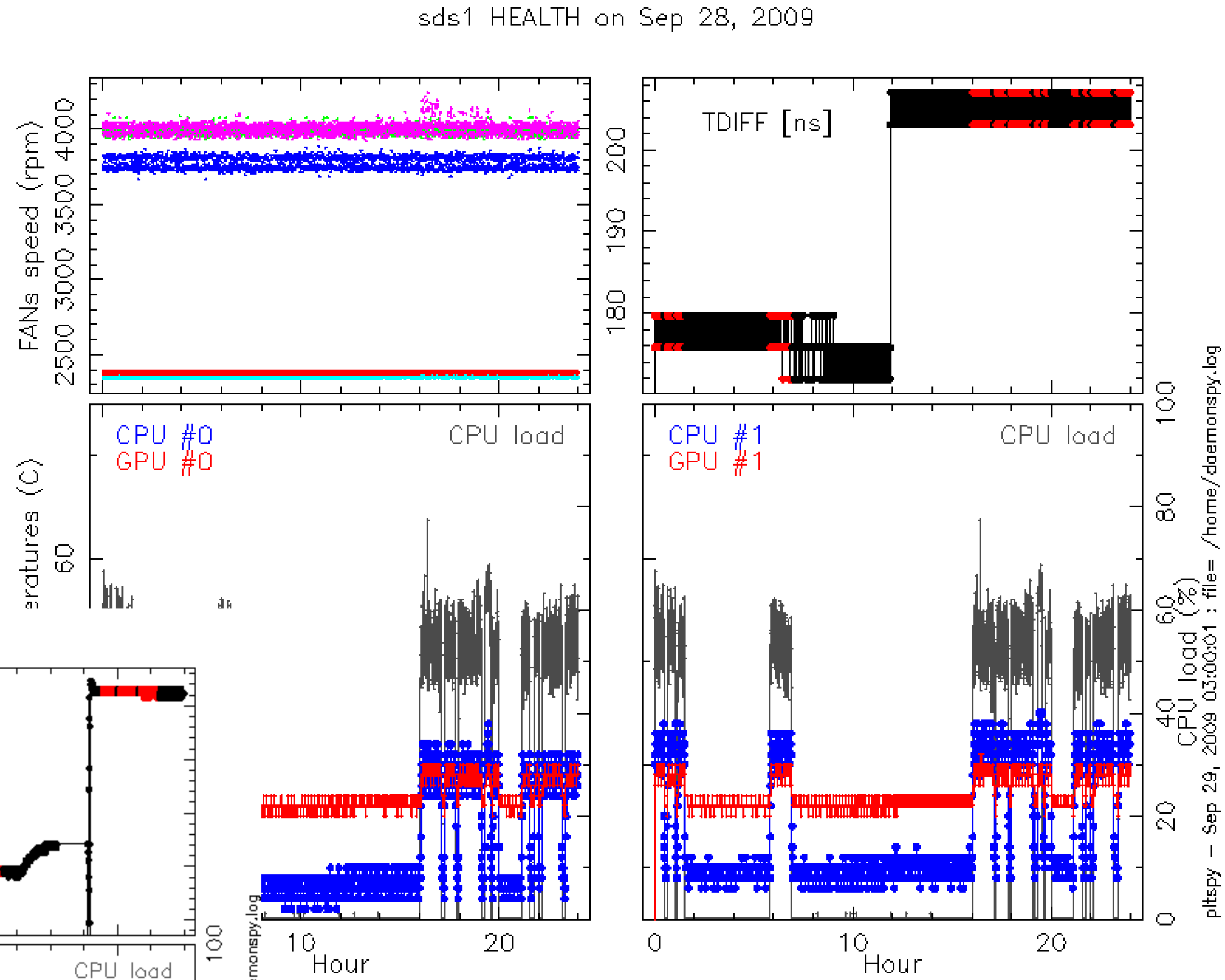


Apr 2008, prototype
without water cooling

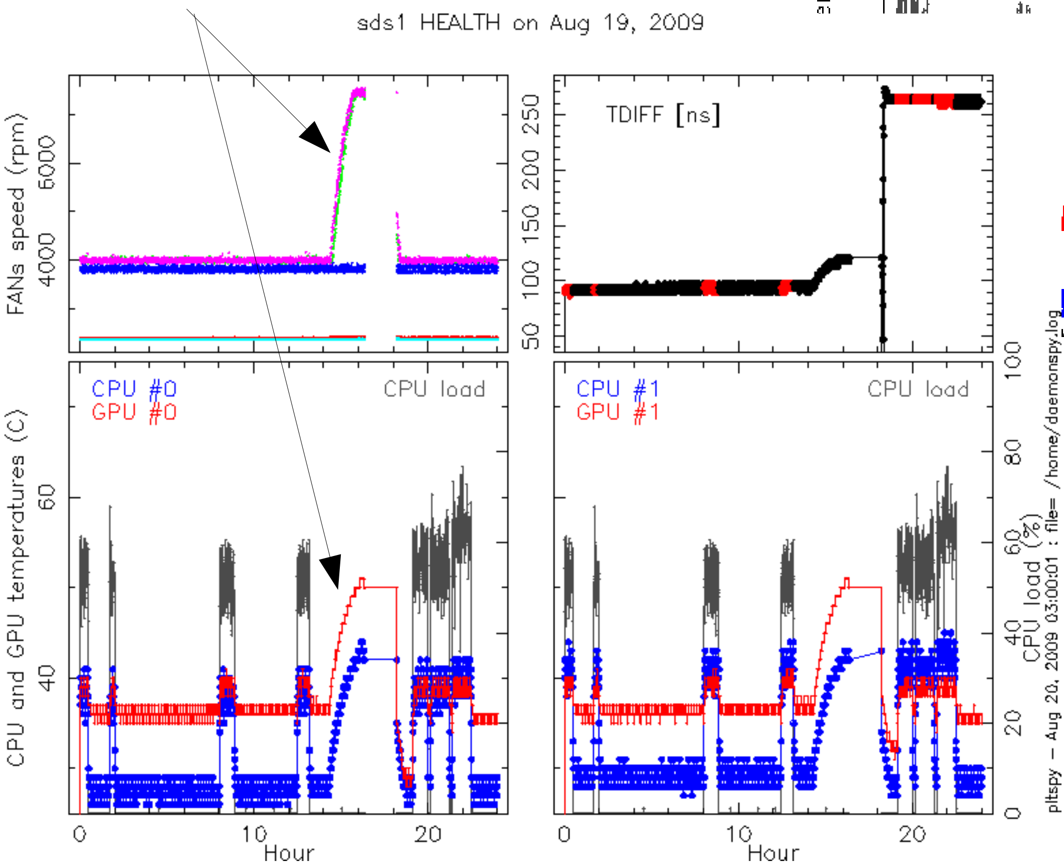
Architecture

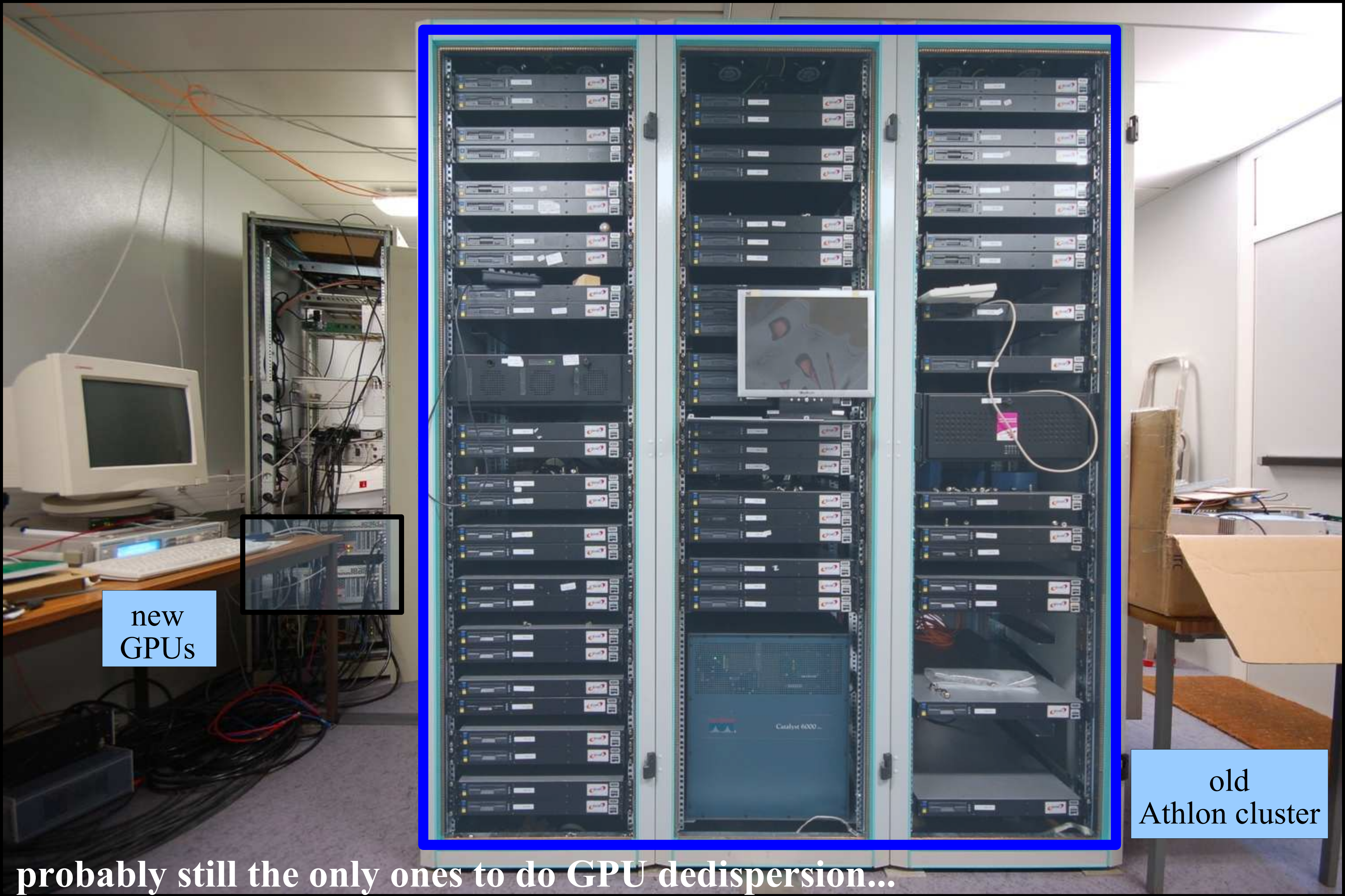


monitoring the system...



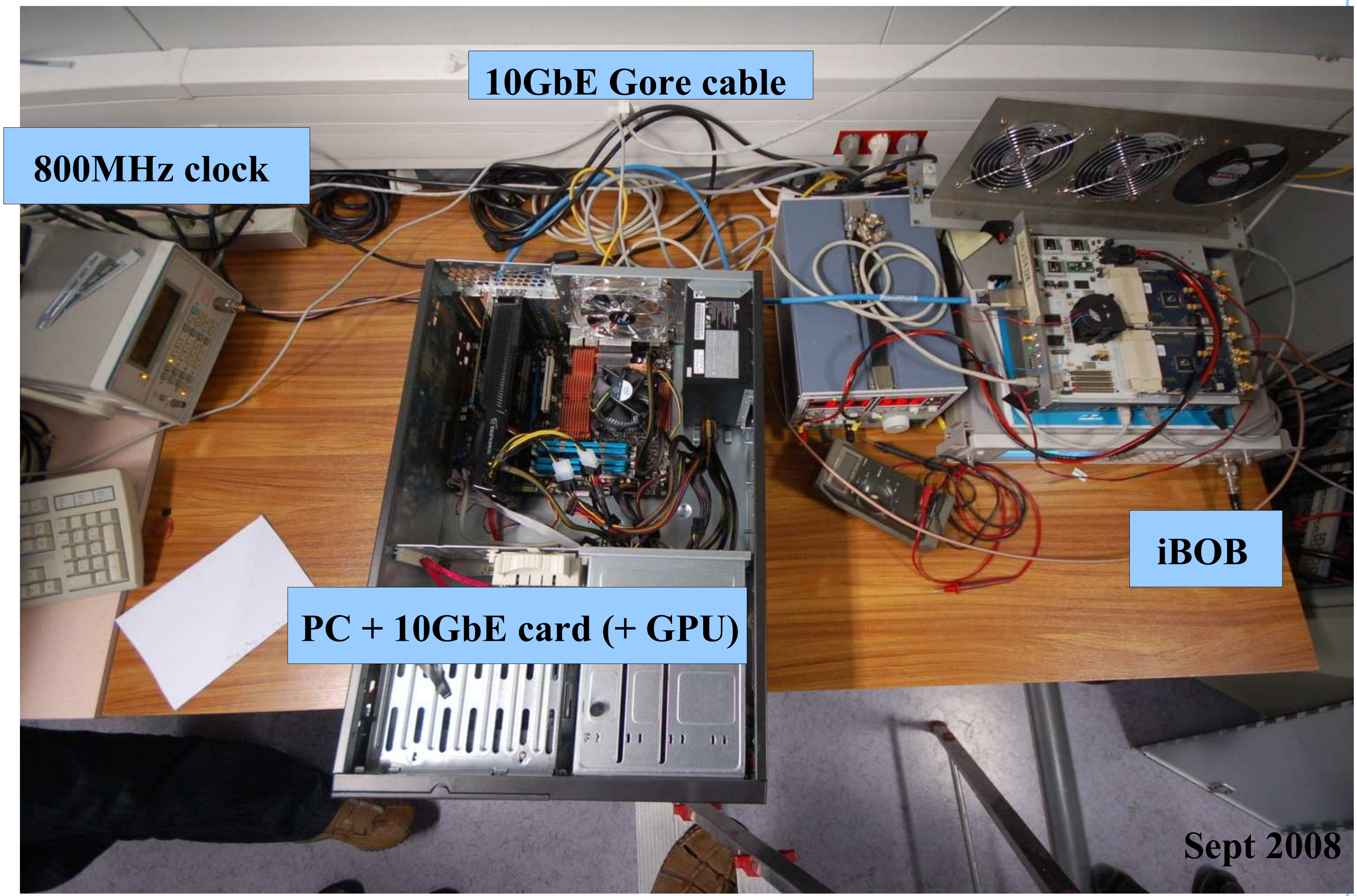
A/C failure



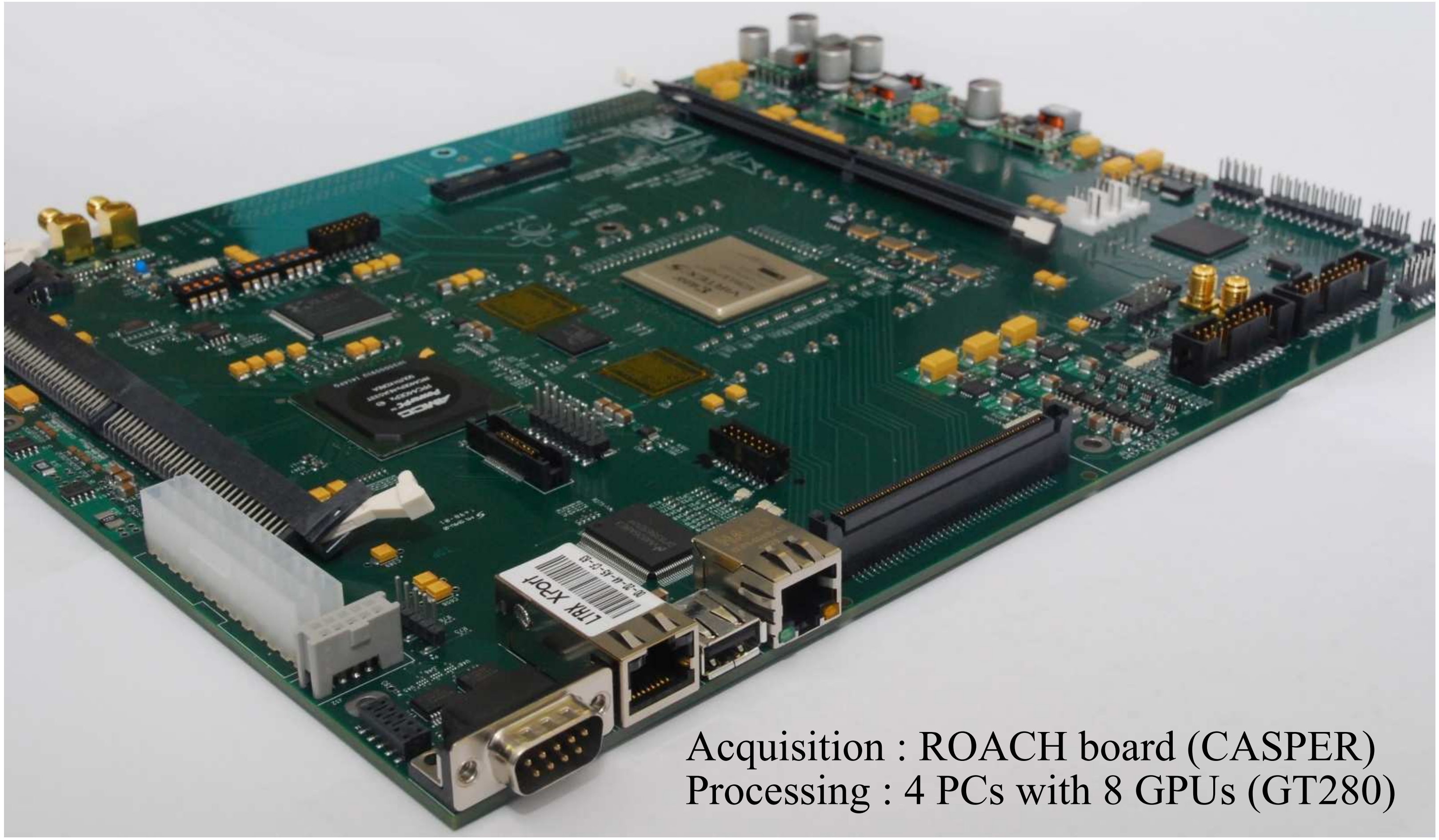

new
GPUs
old
Athlon cluster
probably still the only ones to do GPU dedispersion...

the **next coherent pulsar dedispersor** at Nançay
will have a 400MHz bandwith

iBOB tests conducted on Sept 2008 with P.McMahon



Early 2010 : a 400MHz coherent pulsar dedispersor



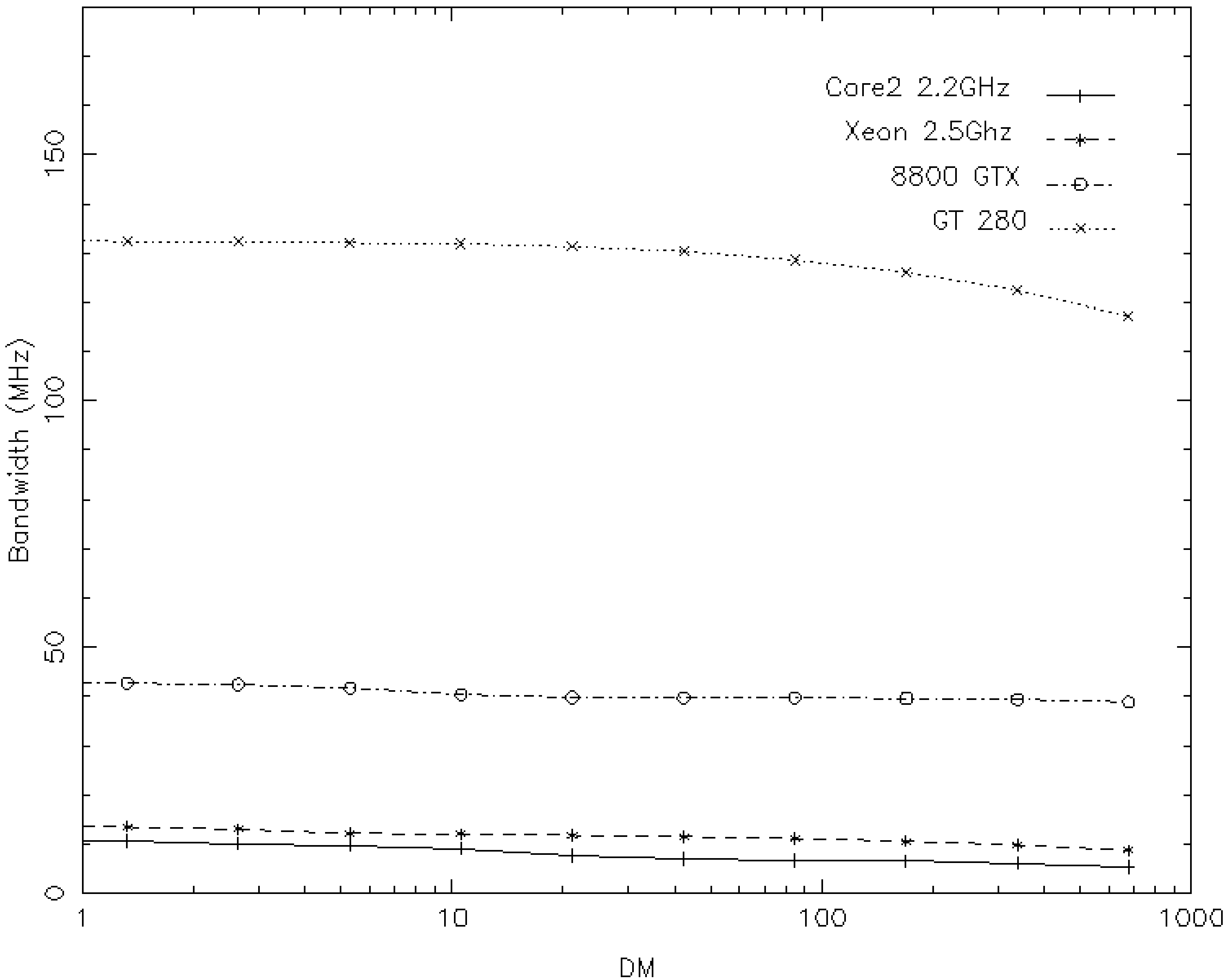
Acquisition : ROACH board (CASPER)
Processing : 4 PCs with 8 GPUs (GT280)

~ 2012-2013

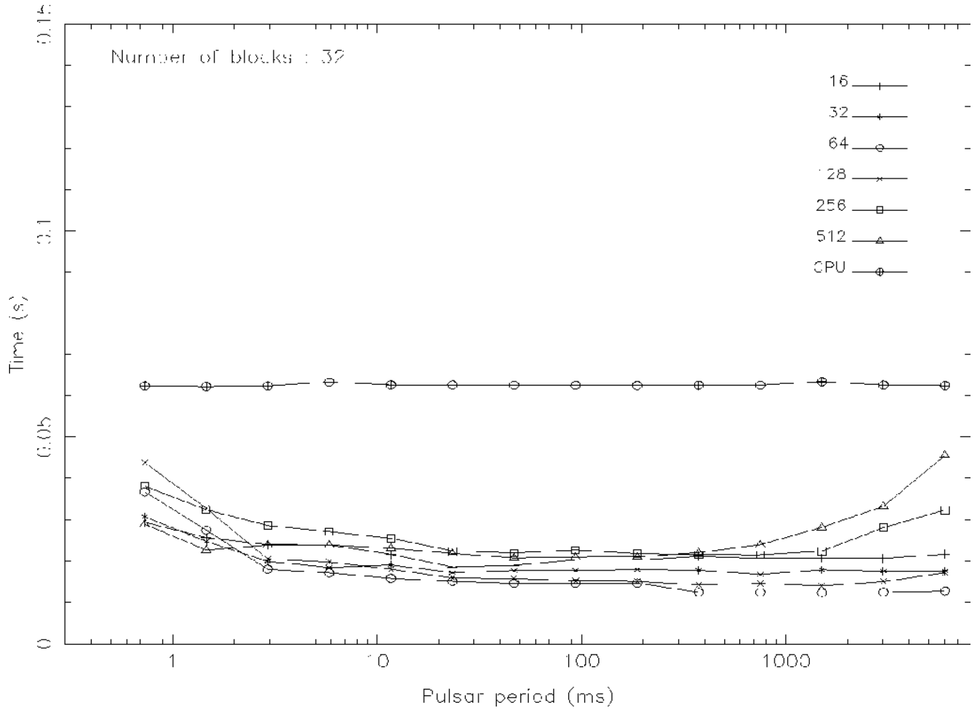
direct sampling of the receivers outputs ?
UNIBOARD collaboration with PRISME Univ Orléans (PhD)
(http://www.radionet-eu.org/uniboard)

the GT200 family

GT280 is **2.5-3 times faster** than the **8800GTX** for dedispersion
(FFT,chirp_mult,FFT^{-1})



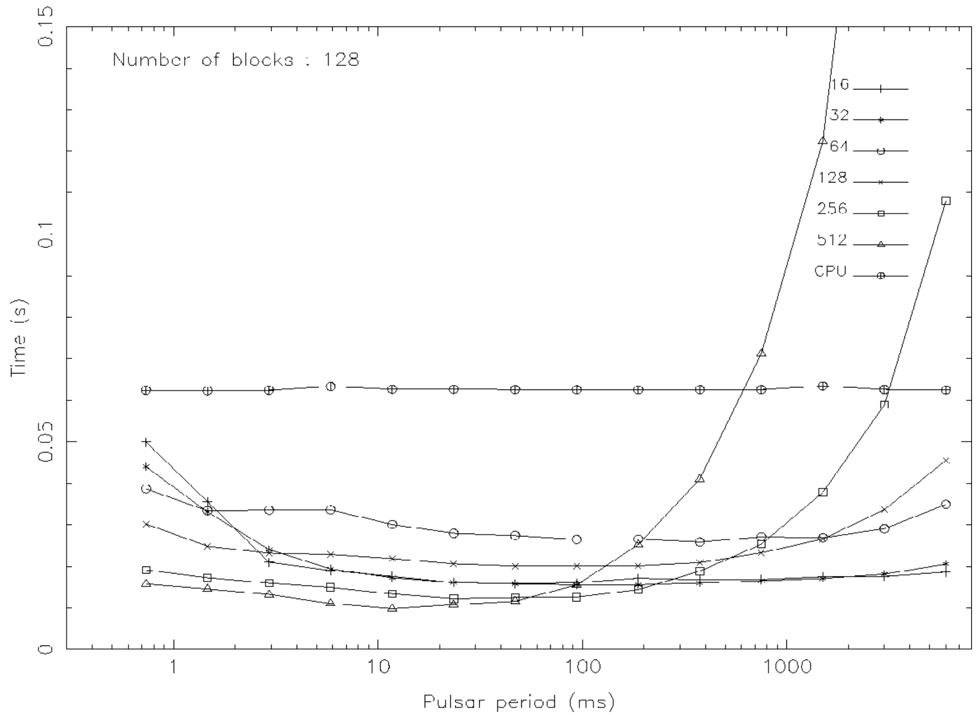
Folding in the GPU



folding

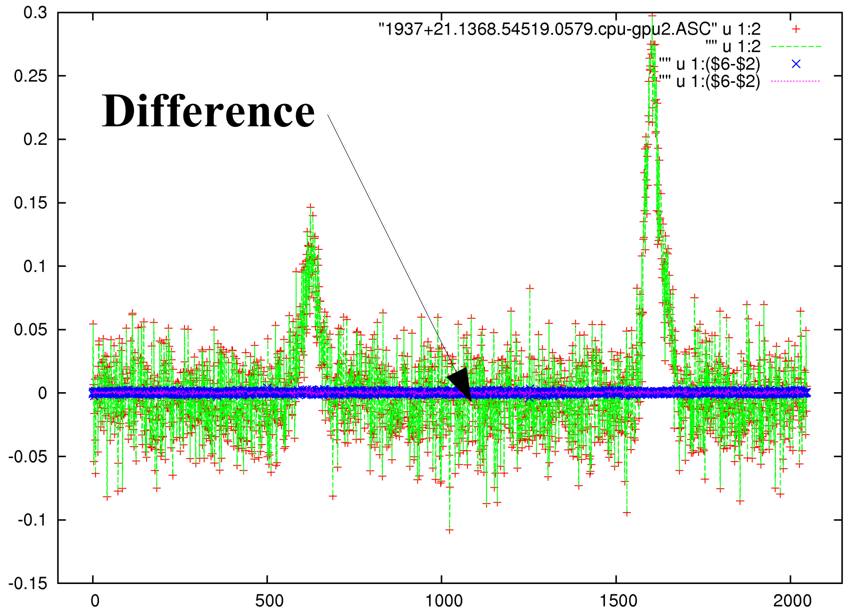
calculate expected rotational phase
for each sample
integrate in a 2k profile array and count
normalize

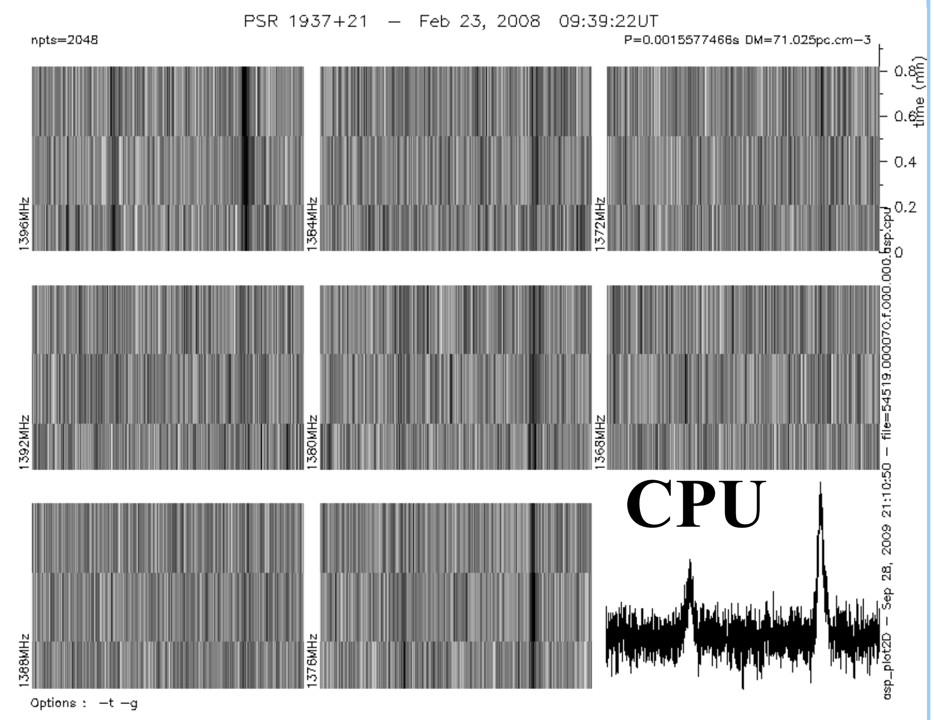
Grégory Desvignes wrote kernel code
and performed tests
for different kernel configurations
(numbers of threads and blocks)

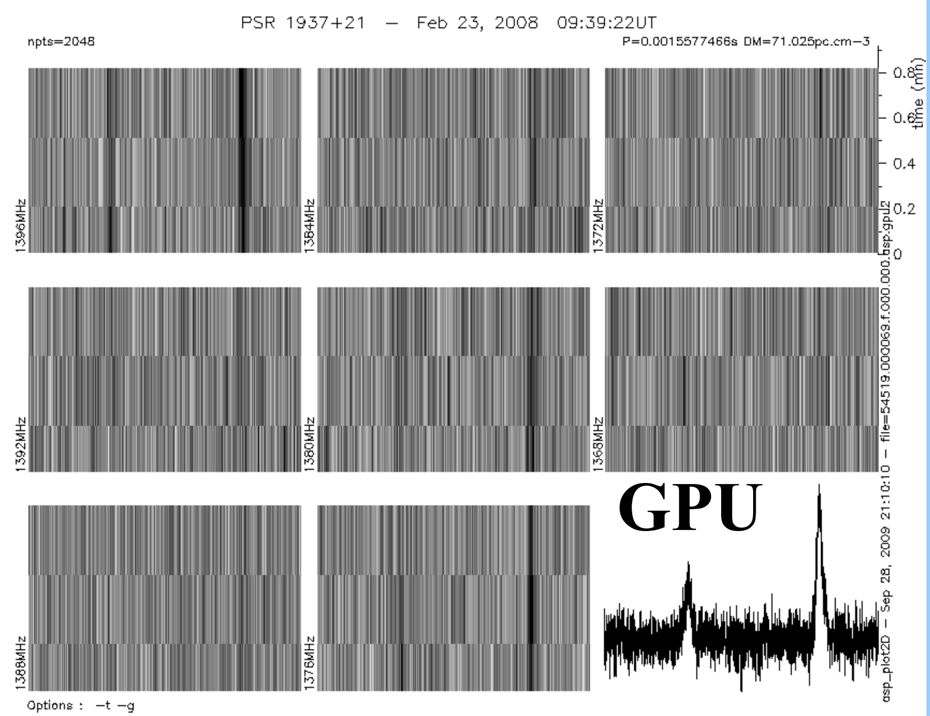


Folding in the GPU

the kernel code to
do the **folding** within the GPU
was tested... successfully !







we are now working on the **re-ordering**
of data received from a PFB
(a matrix transposition)

Conclusion

we guess GPUs are a good alternative
to do coherent dedispersion of pulsars

- they are inexpensive at ~300€ each

- the GT200 generation is able to process
  more than 100MHz bw each

- properly coded, GPUs can do all the job :
  re-ordering data, dedispersing, folding
  (maybe doing everything for bw 100MHz)

for other applications, have a look to
http://www.nvidia.fr/object/cuda_home_fr.html

Post-processing cluster

16 nodes quad-core Intel 2.83GHz
2 disks 1TB RAID 0
Gigabit links

Used to process survey data
- Foster survey 1996-98
- test runs
- future Nançay survey
  - Galactic Plane ?