Enabling Scientific Breakthroughs

Rob van Nieuwpoort

R.vanNieuwpoort@esciencecenter.nl
We work demand-driven
NLeSC eScience competences applied in research

1. Optimized data handling
   Data integration, data base optimization, structured & unstructured data, real time data

2. Big data analytics
   Statistics, machine learning, visualization, text mining

3. Efficient computing
   Distributed & accelerated computing, efficient algorithms

Prevent fragmentation and duplication
## Software

Click on the bars to find software projects.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHN2 point cloud viewer</td>
<td>WebGL point cloud visualization of AHN2</td>
</tr>
<tr>
<td>AMUSE</td>
<td>The Astrophysical Multipurpose Simulation Environment</td>
</tr>
<tr>
<td>CClusTera</td>
<td>A 3D web tool for interactive visualization of hierarchically clustered big data</td>
</tr>
<tr>
<td>Cesiam-ncWMS</td>
<td>3D Globe Visualization of NetCDF data</td>
</tr>
<tr>
<td>Common Sense</td>
<td>User-friendly web application for showing (GIS) data on a map.</td>
</tr>
<tr>
<td>Cross-perspective Topic Modeling</td>
<td>A Gibbs sampler that implements Cross-Perspective Topic Modeling</td>
</tr>
<tr>
<td>DataVaults</td>
<td>Technology of Attachment to a DBMS of large file repositories.</td>
</tr>
<tr>
<td>Differential Evolution</td>
<td>Differential Evolution global optimization algorithm, with Metropolis for uncertainty estimation</td>
</tr>
<tr>
<td>eAstraViz</td>
<td>This tool can convert and visualize radio astronomy measurement sets, as well as most LOFAR intermediate data products. It also does RFI mitigation</td>
</tr>
<tr>
<td>eEcology Annotation Tool</td>
<td>Visualize &amp; annotate GPS measurements of bird movements</td>
</tr>
<tr>
<td>eEcology Tracker calendar</td>
<td>Calendar overview with daily statistics of GPS-tracker</td>
</tr>
<tr>
<td>eWaterLeaf</td>
<td>Web-based visualization for the eWaterCycle project</td>
</tr>
<tr>
<td>ExJS-DateTime</td>
<td>Date/Time form input field for ExJS</td>
</tr>
<tr>
<td>FAIR Data Point</td>
<td>FAIR Data Point Metadata Service</td>
</tr>
<tr>
<td>GoogleEarth Toolbox for MATLAB</td>
<td>Export data from MATLAB to GoogleEarth's KML format.</td>
</tr>
<tr>
<td>Historic Embodied Emotions Model (HEEM) database</td>
<td>275 17th and 18th century Dutch theater texts with HEEM labels</td>
</tr>
<tr>
<td>Kernel Tuner</td>
<td>A simple CUDA/OpenCL kernel tuner in Python</td>
</tr>
</tbody>
</table>

eStep Software used in Projects:

- AHN2 point cloud viewer
- AMUSE
- CClusTera
- Cesiam-ncWMS
- Common Sense
- Cross-perspective Topic Modeling
- DataVaults
- Differential Evolution
- eAstraViz
- eEcology Annotation Tool
- eEcology Tracker calendar
- eWaterLeaf
- ExJS-DateTime
- FAIR Data Point
- GoogleEarth Toolbox for MATLAB
- Historic Embodied Emotions Model (HEEM) database
- Kernel Tuner
- DAA Toolbox for MATLAB
Big Data & Big Compute in Radio Astronomy

Rob van Nieuwpoort
director of technology
Two simultaneous disruptive technologies

• Radio Telescopes
  – New sensor types
  – Distributed sensor networks
  – Scale increase
  – Software telescopes

• Computer architecture
  – Hitting the memory wall
  – Accelerators
Two simultaneous disruptive technologies

- **Radio Telescopes**
  - New sensor types
  - Distributed sensor networks
  - Scale increase
  - Software telescopes

- **Computer architecture**
  - Hitting the memory wall
  - Accelerators
Next-Generation Telescopes: Apertif

Image courtesy Joeri van Leeuwen, ASTRON
LOFAR low-band antennas
LOFAR high-band antennas
Station (150m)
LOFAR: The low-frequency array

- One of the largest telescopes in the world
- ~100,000 omni-directional antennas
- Ten terabit/s, 200 gigabit/s to supercomputer
- Hundreds of teraFLOPS
- 10–250 MHz
- 100x more sensitive

[ John Romein et al, PPoPP, 2014 ]
Think Big Think Huge: The Square Kilometre Array (SKA)

Did you know?
- The SKA will be the world's largest radio telescope.

Did you know?
- The SKA will be so sensitive that it will be able to detect an airport radar on a planet 50 light years away.

Did you know?
- The SKA will use enough optical fibre to wrap twice around the Earth!

Did you know?
- The dishes of the SKA will produce ten times the global internet traffic.

Did you know?
- The aperture arrays in the SKA could produce more than 100 times the global internet traffic.

Did you know?
- The data collected by the SKA in a single day would take nearly two million years to playback on an iPod.

Did you know?
- The SKA central computer will have the processing power of about one hundred million PCs.

Did you know?
- The SKA supercomputer will perform $10^{24}$ operations per second—equivalent to the number of stars in three million Milky Way galaxies—in order to process all the data that the SKA will produce.

Did you know?
- The SKA will contain thousands of antennas with a combined collecting area of about one square kilometre (that's 1 000 000 square metres!).

[ Chris Broekema et al, Journal of Instrumentation, 2015 ]
Science Case

Pulsar Searching
Searching for Pulsars

- **Rapidly rotating neutron stars**
  - Discovered in 1967; ~2500 are known
  - Large mass, precise period, highly magnetized
  - Most neutron stars would be otherwise undetectable with current telescopes

- **“Lab in the sky”**
  - Conditions far beyond laboratories on Earth
  - Investigate interstellar medium, gravitational waves, general relativity
  - Low-frequency spectra, pulse morphologies, pulse energy distributions
  - Physics of the super-dense superfluid present in the neutron star core

Alessio Sclocco, Rob van Nieuwpoort, Henri Bal, Joeri van Leeuwen, Jason Hessels, Marco de Vos

[ A. Sclocco et al, IEEE eScience, 2015 ]
Pulsar Searching Pipeline

- Three unknowns:
  - Location: create many beams on the sky
    [Alessio Sclocco et al, IPDPS, 2012]
  - Dispersion: focusing the camera
    [Alessio Sclocco et al, IPDPS, 2012]
  - Period

- Brute force search across all parameters
- Everything is trivially parallel (or is it?)

- Complication: Radio Frequency Interference (RFI)
  [Rob van Nieuwpoort et al: Exascale Astronomy, 2014]
Challenges

- Application becomes real-time because of the data rates
- Limited window of samples due to memory and compute constraints
  - Only fraction of a second, only limited statistics from the past
  - Only small number of frequency bands
- We can afford only few operations per byte
- Distributed system
  - Information distribution, synchronization, scheduling and load-balancing issues
- Limited power budget
- Investigate best platform, develop new algorithms
Potential of accelerators

• Example: NVIDIA K80 GPU (2014)

• Compared to modern CPU (Intel Haswell, 2014)
  – 28 times faster at 8 times less power per operation
  – 3.5 times less memory bandwidth per operation
  – 105 times less bandwidth per operation including PCI-e

• Compared to BG/p supercomputer
  – 642 times faster at 51 times less power per operation
  – 18 times less memory bandwidth per operation
  – 546 times less bandwidth per operation including PCI-e

• Legacy codes and algorithms are inefficient
• Need different programming methodology and programming models, algorithms, optimizations

• Can we build large-scale scientific instruments with accelerators?
Systems become increasingly hierarchical

- Instruction-level parallelism, vectors, threads, warps, streaming multiprocessors, chips, multiple devices/node, islands, supercomputer, (hierarchical) distributed system

- Need to explicitly address parallelism on each level

- Communication
  - Explicit
  - Overlap communication and computation on all levels
  - Explicit caches, fast local memories, network on-chip, PCI-e, local interconnects, lightpaths

- Even supercomputers are becoming more and more heterogeneous: multiple generations of CPUs and accelerators in one system
Jungle computing with Ibis

- Xenon: middleware-independent deployment
- IPL: portable communication substrate
  - Steaming, malleable, asynchronous, upcalls
  - Solves connectivity issues
  - On top of TCP, UDP, MPI, infiniband, ...
- High-level programming models

[Henri Bal et al, IEEE computer 2010]
Programming hierarchical systems

- Need truly hierarchical programming models
  - Hierarchy-aware MPI (point-to-point and collectives)
  - Example: divide-and-conquer
  - Generic model
  - Proven optimal for shared memory multiprocessors, uniform clusters (Cilk)
  - Shown to work extremely well in hierarchical distributed systems (Satin)
  - Fault-tolerance, malleability, adaptive, speculative parallelism, …
  - [ Rob van Nieuwpoort et al, ACM TOPLAS, 2010 ]

- **Cashmere** integrates Satin & accelerators
  - Mixed programming models
  - Stepwise refinement for performance methodology
  - [ Pieter Hijma et al, IPDPS 2015 ]

- My holy grail: one unified programming model to rule them all
Our Strategy for flexibility, portability

- Investigate algorithms

- OpenCL: platform portability

- Observation type and parameters only known at run time
  - E.g. # frequency channels, # receivers, longest baseline, filter quality, observation type

- Use runtime compilation and auto-tuning
  - Map specific problem instance efficiently to hardware
  - Auto tune platform-specific parameters

- Portability across different instruments, observations, platforms, time!
Histogram: Auto-Tuning Dedispersion on AMD HD7970 for Apertif
Speedup over best possible fixed configuration

Apertif scenario
Pulsar pipeline

SKA1 baseline design, pulsar survey: 2,222 beams; 16,113 DMs; 2,048 periods. Total number of GPUs needed: 140,000. This requires 30 MW. SKA2 should be 100x larger, in the 2023-2030 timeframe.

Apertif and LOFAR: real data
SKA1: simulated data

Speedup over Intel Xeon E5-2620 CPU, 2048x2048 case

Power saving over Intel Xeon E5-2620 CPU, 2048x2048 case
Conclusions

• Exascale changes everything
  – Offline versus streaming, best hardware architecture, algorithms, optimizations
  – All large compute platforms are becoming heterogeneous
  – Needed 8 years to make this work for one application
    We desperately need high-level programming models that incorporate the entire hierarchy!
  – Need new theory as well: from computational complexity to data access complexity

• eScience approach works!
  – Need domain expert for deep understanding & choice of algorithms
  – Need computer scientists for investigating efficient solutions
  – LOFAR has already discovered more than 25 new pulsars!

• Astronomy is a driving force for HPC, Big Data, eScience
  – Techniques are general, already applied in image processing, climate, digital forensics
An example of real time challenges

Investigate algorithms: RFI mitigation
Auto-tuning example, NVIDIA TitanX
Radio Frequency Interference

• RFI is a huge problem for many observations
• Caused by
  – Lightning, vehicles, airplanes, satellites, electrical equipment, GSM, FM Radio, fences, reflection of wind turbines, …
• Best removed offline
  – Complete dataset available
  – Good overview / statistics / model
  – Can spend compute cycles

RFI mitigation
RFI mitigation results

- One robust algorithm for different scales (μs - hours)
  - Filters with exponentially increasing window sizes
- Scalable: linear computational complexity
- Quality almost as good as offline

![Signal to noise ratio chart]

- Non-flagged: 7.06, 7.19
- Threshold: 4.82
- SumThreshold: 5.0
- SumThreshold + Presto rfifind: 78.7
- Flagger: 21.3
- Compute: 78.7
- I/O: 5.0

0% 50% 100%
An example of real time challenges

Auto-tuning: Dedispersion
Dedispersion

[A. Sclocco et al, IPDPS 2014]
[A. Sclocco et al, Astronomy & Computing, 2016]
Auto-tuning platform parameters

Apertif scenario
Auto-tuned performance

Apertif scenario

LOFAR scenario
An example of real time challenges

Changing algorithms: Period search
Period Search: Folding

- Traditional offline approach: FFT
- Big Data requires change in algorithm: must be real time & streaming

Stream of samples

Period 8:

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ...
```

Period 4:

```
0 1 2 3 +
4 5 6 7 +
8 9 10 11 +
12 13 14 15 +
```

[A. Sclocco et al, IEEE eScience, 2015]
Optimizing Folding

- Build a tree of periods to maximize reuse
- Data reuse: walk the paths from leaves to root
Pulsar B1919+21 in the Vulpecula nebula.
Pulse profile created with real-time RFI mitigation and folding, LOFAR.

Background picture courtesy European Southern Observatory.
Today’s discovery

- Millisecond pulsar
  PSR J1552+54
- Discovered at 135 MHz
- Lowest observing frequency an MSP has been discovered.
- Non-detections at 1400MHz by Lovell and Nancay.
- Use of LOFAR indispensable.
Credits

Rob van Nieuwpoort
Alessio Sclocco
Pieter Hijma
Chris Broekema
Ana Lucia Varbanescu
John Romein
Ger van Diepen
Joeri van Leeuwen
Jason Hessels
Souley Madougou
Tim Cornwell
Bruce Almegreen
Henri Bal
Henk Sips

and many others!
Next-Generation Telescopes: Apertif
Backup slides
Pulsar pipeline Performance Breakdown

- **HD7970**
- **K20**
- **Xeon Phi**

- **Apertif**
- **LOFAR**
- **SKA 1**

- **Period search**
- **Dedispersion**
- **I/O**
Auto-tuning example

Beam Forming, GTX580
Pulsar pipeline (real time)

- Pulsar pipeline
- Poly-Phase Filter
- RFI mitigation
- Beam Former
- Dedispersion
- Period search
- Signal-to-Noise

Graphical representation of the pipeline processes.
Modern Computer Architectures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak (gflops)</td>
<td>13.6</td>
<td>307</td>
<td>8736</td>
</tr>
<tr>
<td>Memory bandwidth (GByte/s)</td>
<td>13.6</td>
<td>59</td>
<td>480</td>
</tr>
<tr>
<td>Operations / byte</td>
<td>1.0</td>
<td>5.2</td>
<td>18.2</td>
</tr>
</tbody>
</table>
## Modern Computer Architectures

<table>
<thead>
<tr>
<th></th>
<th>IBM BG/p</th>
<th>CPU Haswell (2015)</th>
<th>GPU NVIDIA K80 (2014)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak (gflops)</td>
<td>13.6</td>
<td>307</td>
<td>8736 (28x CPU)</td>
</tr>
<tr>
<td>Memory bandwidth (GByte/s)</td>
<td>13.6</td>
<td>59</td>
<td>480 (8x CPU)</td>
</tr>
<tr>
<td>Operations / byte</td>
<td>1.0</td>
<td>5.2</td>
<td>18.2 (3.5x CPU)</td>
</tr>
</tbody>
</table>

Including data movement (PCI-e): 546 (105x CPU)
## Modern Computer Architectures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak (gflops)</td>
<td>13.6</td>
<td>307</td>
<td>8736</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(28x CPU)</td>
</tr>
<tr>
<td>Memory bandwidth (GByte/s)</td>
<td>13.6</td>
<td>59</td>
<td>480</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(8x CPU)</td>
</tr>
<tr>
<td>Operations / byte</td>
<td>1.0</td>
<td>5.2</td>
<td>18.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(3.5x CPU)</td>
</tr>
</tbody>
</table>

**Including data movement (PCI-e):** 546 (105x CPU)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>gflops / Watt</td>
<td>0.57</td>
<td>3.61</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(8.1x CPU)</td>
</tr>
</tbody>
</table>
### Modern Computer Architectures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak (gflops)</td>
<td>13.6</td>
<td>307</td>
<td>8736</td>
</tr>
<tr>
<td>Memory bandwidth (GByte/s)</td>
<td>13.6</td>
<td>59</td>
<td>480</td>
</tr>
<tr>
<td>Operations / byte</td>
<td>1.0</td>
<td>5.2</td>
<td>18.2</td>
</tr>
</tbody>
</table>

**Including data movement (PCI-e):** 546 (105x CPU)

gflops / Watt: 0.57, 3.61, 29.1 (8.1x CPU)

- Huge performance potential and increase in power efficiency
- Legacy codes are inefficient on modern architectures
- Need completely different optimizations, algorithms, programming models
- Can we build large-scale scientific instruments with accelerators?
Big Data == Big Compute

• We need “Big Compute” for processing Big Data
  – Currently petaflops
  – SKA will be exascale
    [Chris Broekema et al, Journal of Instrumentation, 2015]

• Large-scale parallelism

• Accelerators
  – GPUs NVIDIA, AMD; Intel Xeon Phi,
    FPGAs, ASICs, DSPs, …
## Modern Computer Architectures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak (gflops)</td>
<td>13.6</td>
<td>85</td>
<td>307</td>
<td>4500</td>
<td>8736</td>
</tr>
<tr>
<td>Memory bandwidth (GByte/s)</td>
<td>13.6</td>
<td>25.6</td>
<td>59</td>
<td>288</td>
<td>480</td>
</tr>
<tr>
<td>Operations / byte</td>
<td>1.0</td>
<td>3.3</td>
<td>5.2</td>
<td>15.6</td>
<td>18.2</td>
</tr>
</tbody>
</table>
# Modern Computer Architectures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak (gflops)</td>
<td>13.6</td>
<td>85</td>
<td>307</td>
<td>4500</td>
<td>8736</td>
</tr>
<tr>
<td>Memory bandwidth (GByte/s)</td>
<td>13.6</td>
<td>25.6</td>
<td>59</td>
<td>288</td>
<td>480</td>
</tr>
<tr>
<td>Operations / byte</td>
<td>1.0</td>
<td>3.3</td>
<td>5.2</td>
<td>15.6</td>
<td>18.2</td>
</tr>
</tbody>
</table>

Including data movement (PCI-e): 546 (105x CPU)
### Modern Computer Architectures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak (gflops)</td>
<td>13.6</td>
<td>85</td>
<td>307</td>
<td>4500</td>
<td>8736</td>
</tr>
<tr>
<td>Memory bandwidth (GByte/s)</td>
<td>13.6</td>
<td>25.6</td>
<td>59</td>
<td>288</td>
<td>480</td>
</tr>
<tr>
<td>Operations / byte</td>
<td>1.0</td>
<td>3.3</td>
<td>5.2</td>
<td>15.6</td>
<td>18.2</td>
</tr>
<tr>
<td><strong>Including data movement (PCI-e):</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>546 (105x CPU)</strong></td>
<td></td>
</tr>
<tr>
<td>gflops / Watt</td>
<td>0.57</td>
<td>0.65</td>
<td>3.61</td>
<td>18.0</td>
<td>29.1</td>
</tr>
</tbody>
</table>
### Modern Computer Architectures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak (gflops)</td>
<td>13.6</td>
<td>85</td>
<td>307</td>
<td>4500</td>
<td>8736</td>
</tr>
<tr>
<td>Memory bandwidth (GByte/s)</td>
<td>13.6</td>
<td>25.6</td>
<td>59</td>
<td>288</td>
<td>480</td>
</tr>
<tr>
<td>Operations / byte</td>
<td>1.0</td>
<td>3.3</td>
<td>5.2</td>
<td>15.6</td>
<td>18.2</td>
</tr>
</tbody>
</table>

Including data movement (PCI-e): **546 (105x CPU)**

gflops / Watt

|                         | 0.57           | 0.65               | 3.61               | 18.0                        | 29.1                   | (8.1x) CPU            |

- Legacy codes are inefficient on modern architectures
- Need completely different optimizations, algorithms, programming models
- Can we build large-scale scientific instruments with accelerators?
**My Interests: Efficient Computing**

- **Generic** hierarchical programming models
- Efficiently mapping *challenging* scientific applications to these *complex* platforms
  - Performance
  - Power
  - Programmability
- This talk: example from astronomy
Big Data in Astronomy

- Start of the pipeline: huge volume, structured, 99.999% noise
- Intermediate: huge²
- Final product
  - Can be 1 bit (pulsar)
  - Can be image cubes: 2D sky, frequency, time
  - Can be source catalogs
- Complexity of data and algorithms increases
<table>
<thead>
<tr>
<th>SKA1 details</th>
<th>SKA1 mid</th>
<th>SKA1 low</th>
<th>SKA1 survey</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of receivers</td>
<td>254 (190 + 64)</td>
<td>262,144 (1024 x 256)</td>
<td>96 (64 + 36)</td>
</tr>
<tr>
<td>Receiver diameter</td>
<td>15 m (13.5 m)</td>
<td>35 m</td>
<td>15 m (12 m)</td>
</tr>
<tr>
<td>Maximum baseline</td>
<td>100 km</td>
<td>70 km</td>
<td>50 km</td>
</tr>
<tr>
<td>Input bandwidth</td>
<td>34 Tbps</td>
<td>73 Tbps</td>
<td>47 Tbps</td>
</tr>
<tr>
<td>Red’q Compute capacity</td>
<td>52 PFLOPS</td>
<td>25 PFLOPS</td>
<td>72 PFLOPS</td>
</tr>
</tbody>
</table>
Imaging pipeline

Real-time

Antenna → Light paths to correlator → visibilities

offline

RFI mitigation → Calibration → Gridding → Source finder

Flag Mask → visibilities → catalog
Imaging pipeline: scaling up

Real-time

Antenna → Light paths to correlator → RFI mitigation → Calibration → Gridding

Catalog

Offline

Source finder
Data distribution

- SKA1: construction 2018-2023; early science 2020+
- SKA2: construction 2023 - 2030
- SKA is distributed instrument by design
  - Western Australia and South Africa
  - Central archive?
  - Replicate?
- Distribute image cubes to SKA data science centers
  - Image cubes can be large: ~ 20K x 20K x 1K x double
  - Rough estimate: 100 Gbit/s
- Infrastructure?
- Bring processing to the data?
Flexibility and Portability

- Many different instruments
- Many different science cases, observation types, and parameters
- Life time of an instrument is much longer than life time of compute hardware
The balance has shifted

- Legacy codes are inefficient on modern architectures
  - Need completely different optimization, algorithms

- Arithmetic Intensity:
  - \( O(1) \):
    - SpMV, BLAS1,2
  - \( O(\log(N)) \):
    - Stencils (PDEs)
    - Lattice Methods
  - \( O(N) \):
    - FFTs
    - Dense Linear Algebra (BLAS3)
    - Particle Methods
CPU versus GPU Compute Performance
CPU versus GPU memory performance
Supercomputers & Accelerators
Data reuse
Auto-Tuning Dedispersion

Apertif

LOFAR
GPU Pulsar pipeline schematic

- **Dedispersion**
- **Transpose**
- **SNR Dedispersed**
- **Folding**
- **SNR Folded**

**Input Data** → **Dedispersed Data** → **Transposed Data** → **Dedispersed SNR Data** → **Folded Data** → **Folded SNR Data**

- **Single peaks** (Fast Radio Bursts)
- **Pulse profiles**
- **Candidate list**
A Real-Time Radio Transient Pipeline for ARTS

ARTS, the Apertif Radio Transient System, is the system Astron is building to find FRBs in real-time.

Fast Radio Burst (FRB): “high energy astrophysical phenomenon manifested as a transient radio pulse lasting only a few milliseconds” (Wikipedia)
Data reuse

• Data reuse

• Automatically optimize for occupancy
  – (keep compute cores busy)

• Automatically optimize for memory bandwidth
OpenCL: The Khronos group
OpenCL: Open Compute Language

- Architecture independent
- Explicit support for many-cores
- Low-level host API
  - Uses C library, no language extensions
- Separate high-level kernel language
  - Explicit support for vectorization
eAstronomy peer reviewed Publications

   *The LOFAR Transients Pipeline*

2. Alessio Sclocco, Henri E. Bal, Rob V. van Nieuwpoort
   *Finding Pulsars in Real-Time.*
   11th IEEE International Conference on eScience, 31 August - 4 September, 2015, Munich, Germany.

3. Alessio Sclocco, Henri E. Bal, Jason Hessels, Joeri van Leeuwen, Rob V. van Nieuwpoort.
   *Auto-Tuning Dedispersion for Many-Core Accelerators.*
   28th IEEE International Parallel & Distributed Processing Symposium (IPDPS), May 19-23, 2014, Phoenix (Arizona), USA.

   *Real-Time Pulsars Pipeline Using Many-Cores.*
   AAS Exascale Radio Astronomy Meeting, 30 March - 4 April, 2014, Monterey (California), USA.

5. Rob V. van Nieuwpoort and the LOFAR team:
   *Exascale Real-Time Radio Frequency Interference Mitigation.*

6. Alessio Sclocco, Rob V. van Nieuwpoort.
   *Pulsar Searching with Many-Cores.*

7. Alessio Sclocco, Ana Lucia Varbanescu, Jan David Mol, Rob V. van Nieuwpoort.
   *Radio Astronomy Beam Forming on Many-Core Architectures.*
   26th IEEE International Parallel & Distributed Processing Symposium (IPDPS), May 21-25, 2012, Shanghai, China.

8. Alessio Sclocco, Joeri van Leeuwen, Henri E. Bal, Rob V. van Nieuwpoort:
   *A Real-Time Radio Transient Pipeline for ARTS.*

9. P. Chris Broekema, Rob V. van Nieuwpoort and Henri E. Bal:
   *The Square Kilometre Array Science Data Processor Preliminary Compute Platform Design.*

10. Alessio Sclocco, Joeri van Leeuwen, Henri E. Bal, Rob V. van Nieuwpoort:
    *Real-Time Dedispersion for Fast Radio Transient Surveys, using Auto Tuning on Many-Core Accelerators.*
    accepted for publication in Astronomy and Computing, 2016
Masters theses

• Rene van Klink:  
  In progress  
  Working title: Auto-tuning memory layouts  
  Vrije Universiteit Amsterdam,  
  Netherlands eScience Center, 2016.

• Linus Schoemaker:  
  Removing Radio Frequency Interference in the LOFAR using GPUs.  
  Vrije Universiteit Amsterdam,  
  Netherlands eScience Center, 2015.

• Jan Kis:  
  Auto-tuning a LOFAR radio astronomy pipeline in JavaCL.  
  Vrije Universiteit Amsterdam,  
  Netherlands eScience Center, 2013.
## Grants / “spinoffs”

<table>
<thead>
<tr>
<th>PI</th>
<th>Title</th>
<th>Call</th>
<th>Amount (K euro)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Martin Kersten</td>
<td>Big data for the big bang</td>
<td>Dome</td>
<td>200</td>
</tr>
<tr>
<td>Martin Kersten</td>
<td>Compressing the sky</td>
<td>NLeSC</td>
<td>50</td>
</tr>
<tr>
<td>Joeri van Leeuwen</td>
<td>ARTS — the Apertif Radio Transient System</td>
<td>NOVA</td>
<td>730</td>
</tr>
<tr>
<td>Joeri van Leeuwen</td>
<td>ARTS — the Apertif Radio Transient System</td>
<td>NWO-M</td>
<td>590 + 540 matching</td>
</tr>
<tr>
<td>John Romein</td>
<td>Radio-Telescope Algorithms for Many-Core Processor Architectures</td>
<td>NWO Open Competitie</td>
<td>140</td>
</tr>
</tbody>
</table>
Three LOFAR pipelines

- Antenna
- Light paths to central processor
- Imaging mode
- Source finder
- Catalog
- Beam forming mode
- Pulsar pipeline
- Transient pipeline: imaging in real-time
RFI GPU Results

- Ported to GPUs
- Up to 200 LOFAR stations in real time on a single GPU
127-beam tied-array observation using the LOFAR Superterp

Cumulative S/N of PSR B2217+47 in 127 Simultaneous Tied-Array Beams

PSR B2217+47

Courtesy LOFAR Transients Key Project