

ngVLA

Hybrid Correlators and Future Directions

Jonathon Kocz

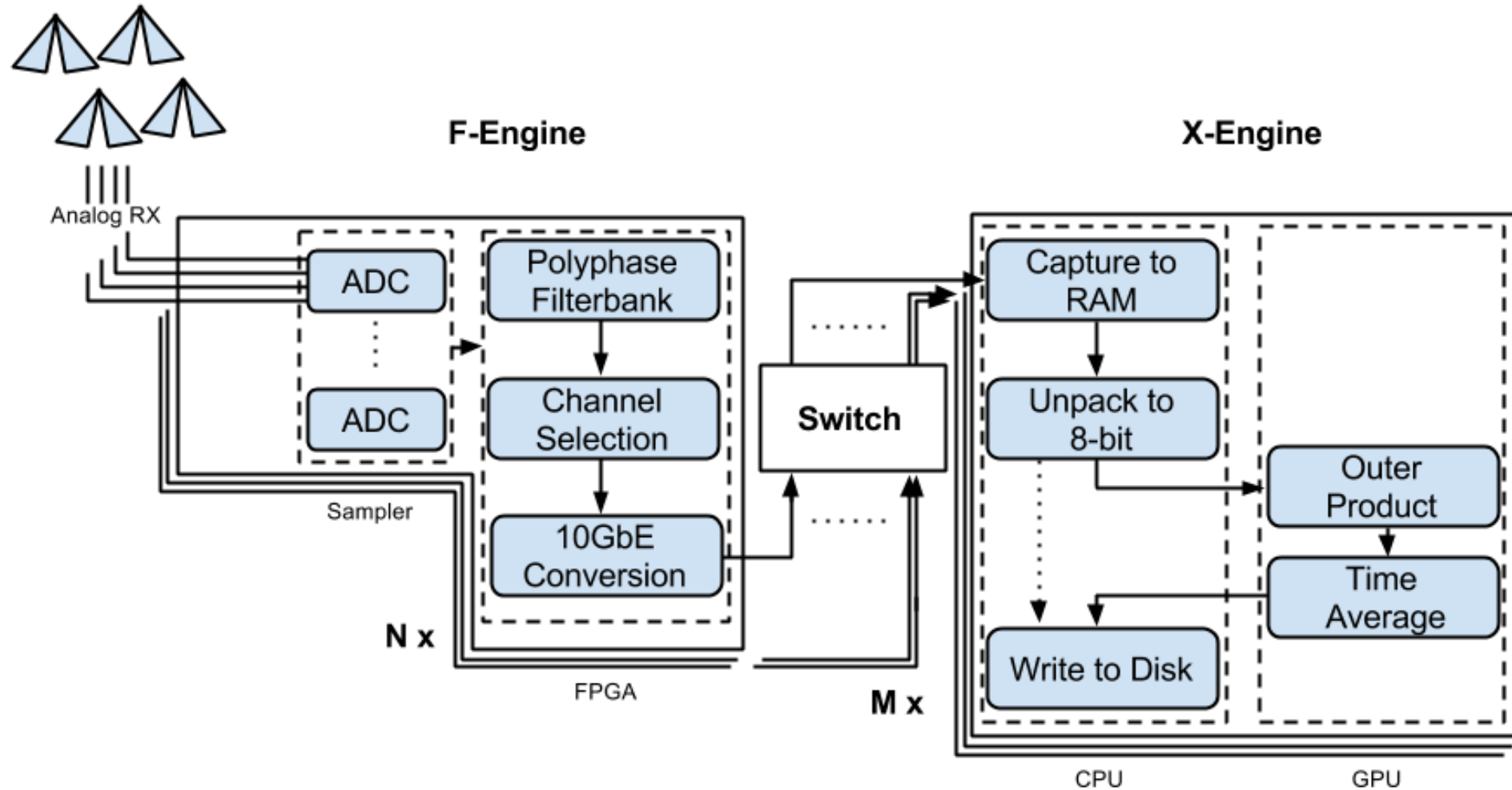
"When you build these things that cost a million dollars it's a good idea to have a meeting to make sure you don't screw up" D. Werthimer

Who is using a “hybrid”?



- So far mostly low frequency, low bandwidth.

LEDA-512



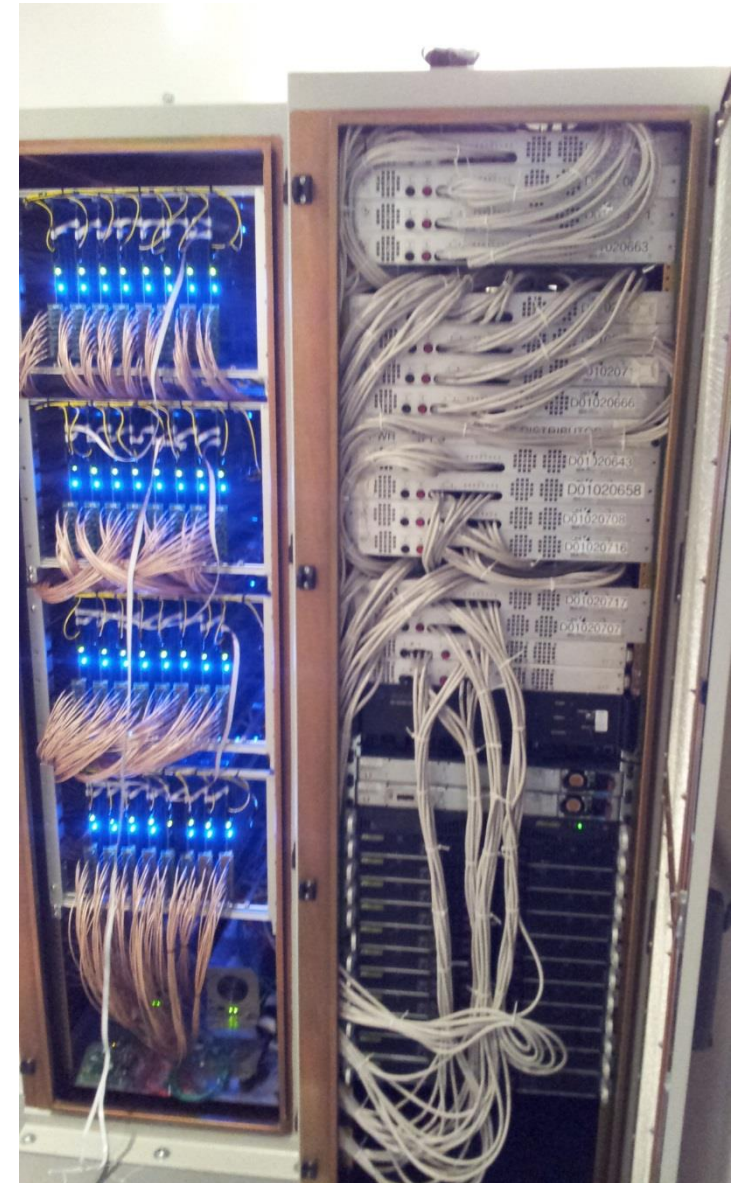
Resources

Hardware:

- 32x 16-input ADCs
 - 16x ROACH2
 - PFB (4096 channels)
 - Channel selection (2398 channels)
 - Packetization
 - Switch (235.4 Gb/s)
 - 11x Dual CPU servers
 - Dual 8-core
 - Capture 21.4Gbps
 - Format for GPU
 - 22x K20X GPUs
 - Cross-multiplication
 - Time averaging
- **Power consumption @ Peak:**
 - FPGA/Switch:
 - 9.5A
 - 1648W
 - GPU servers:
 - 35A (Estimated)
 - 7370W
 - Total estimated:
 - 44.5A
 - 9018W

<http://psrdada.sourceforge.net>

Clark, La Plante, Greenhill, JHPC <https://github.com/GPU-correlators/xGPU>

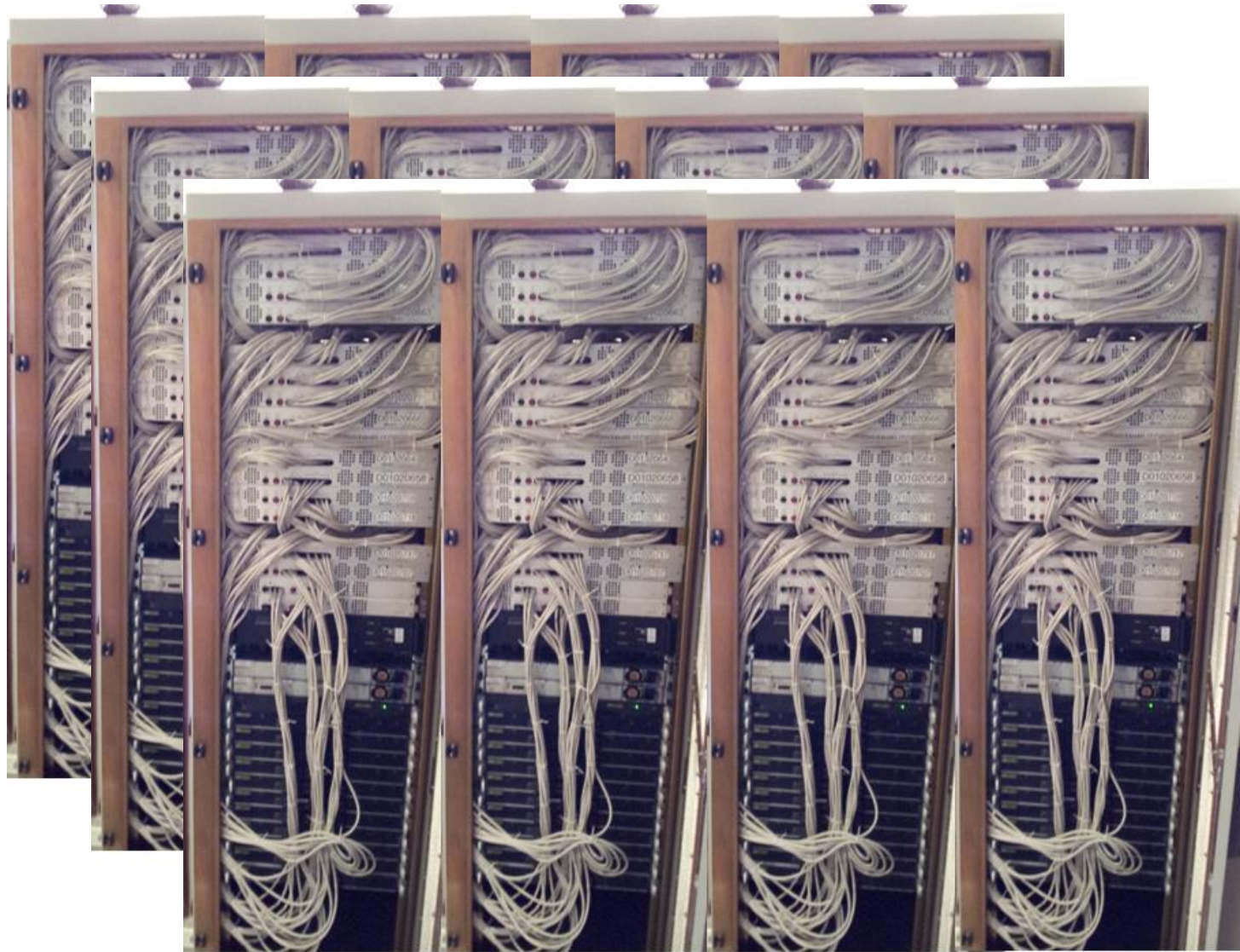


Pros/Cons:

- Deployment time:
 - LEDA-32 August 2012: 36 hrs
 - LEDA-64 June 2013 : 3 days
 - LEDA-512 August 2013: 5 days
- Flexible: Adding pulsar gating, beam former. Incremental development.
- Can reconfigure to do data processing on the nodes.
- Power.
- Physical space
- Not just power/device space. Networking becomes a big issue. Longest part of LEDA development was getting data into the computer node reliably.
- Data transfer: We maxed out at about 30Gbps / node, which concerned me, but I know that CHIME at least is now up to 50Gbps / node.

Limitations:

- How would this scale to ngVLA?
- Computationally fine. Power and space would be a problem!
- $N=256$, $BW = 50\text{GHz}$,
 $\text{TCMAC}=6550$
- $\text{ngVLA} = 436 \times \text{L512}$
- However: $\text{ngVLA} = 262 \times \text{CHIME}$



Variations on a theme:

- MeerKAT using Tegra K1/X boards to develop “Ironhive”. (See Simon Ratcliffe talk at GTC2015)

Tegra X1	Tesla K40
1056 x Tegra X1	2x K40, 2xCPU
\$350k	\$1,056k
12.4kW	57.5kW



Musings:

- The correlator must be viewed as a complete system: from ADC to end data product.
- All technologies are moving forward.
 - Larry's work on a low power correlator chip.
 - Already at 20GHz bandwidth in current FPGA - F.
 - GPUs are becoming more power efficient generally, but also increasing in software design techniques. Downside of host system overhead is also an upside in allowing latest GPUs to be inserted in the host without hardware development time required.
- We need a life cycle cost function, not just initial development: Development time, hardware cost, power costs, upgrade costs.
 - Power calculations for GPUs/FPGAs are still scary compared to ASIC. The only question is, given these numbers, could a FPGA/GPU solution still be more cost effective than an ASIC solution? – I don't know the answer to that question! (Why we need the cost function!).

