BIG COMPUTE: UNDER THE HOOD

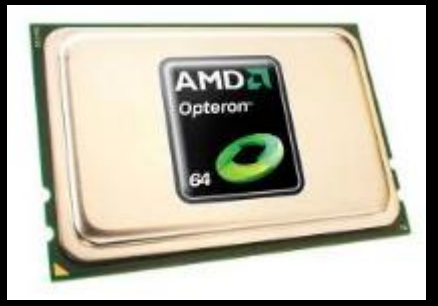
ALAN LEE

JANUARY 31, 2017

OUTLINE

- Introductory remarks
- Definitions and basics
- Example 1: The Square Kilometer Array (SKA)
- Example 2: Whole Genome Sequencing (WGS)
- Under the hood

INTRODUCTORY REMARKS



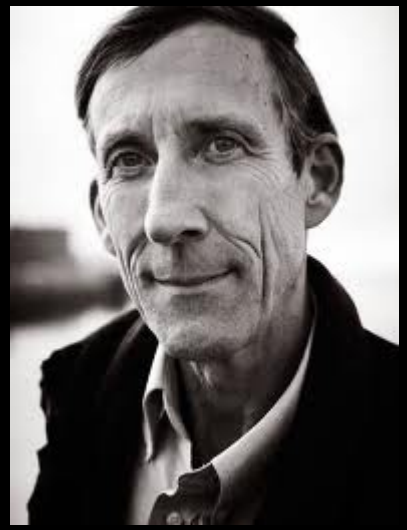
AMD powers millions of the world’s personal computers, tablets, game consoles, embedded devices, and cloud servers.

AMD

*“**Big data** is a collection of datasets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.”*

- Wikipedia

http://en.wikipedia.org/wiki/Big_data



"Big data is what happened when the cost of storing information became less than the cost of ***making the decision*** to throw it away.“

-George Dyson

http://longnow.org/seminars/02013/mar/19/no-time-there-digital-universe-and-why-things-appear-be-speeding/

SO WHAT IS BIG COMPUTE?

AMD

Glib answer: Performing large calculations on Big Data

Note: This is not a great answer because one can apply Big Compute to relatively small data sets.

A better answer: The use of massive, often parallel, computation to solve previously intractable problems

DEFINITIONS AND BASICS

ARCHITECTURE VS MICROARCHITECTURE

AMD

- Architecture:

  Describes the high level attributes of the system. Sometimes referred to as Instruction Set Architecture when applied to processors. Examples include x86 and ARM.

  - For the purist, Architecture consists of instructions, data types, and addressing modes
  - A programmer can “see” the architecture

- Microarchitecture:

  Describes the implementation details of the processor. Examples include pipelining and instruction level parallelism.

  - The microarchitecture is most often hidden from programming languages, but it is often quite important to the programmer

Note: These terms are often conflated in casual discussions with computer engineers

BASIC CHIP DEFINITIONS

AMD

- CPU: Central Processing Unit. Current CPUs include multiple cores, on-package I/O and sometimes include integrated GPUs for display and compute purposes.
- GPU: Graphics Processing Unit. Current GPUs are comprised of many small compute units that are optimized for parallel operations. Specialized graphics hardware is often included to support graphics and displays.
- FPGA: Field Programmable Gate Array. The FPGA is made up of a large number of logic blocks surrounded by a digital routing fabric. Current FPGAs often include floating point building block hardware, small CPU cores and dedicated I/O circuitry such as memory controllers.
- DSP: Digital Signal Processor. DSPs include a small number of cores that are highly optimized for multiply-accumulate operations and other operations often used in signal processing algorithms. DSPs include large amounts of tightly integrated signal processing I/O.
- ASIC: Application Specific Integrated Circuit. ASICs are specialized circuitry implemented as a chip for a specific purpose.

COMPUTING DEVICES COMPARISON

AMD

Device	Ease of Programmability	Application Flexibility	Floating Point Capability	Energy Efficiency
CPU	Easy	High	100's GFLOPS range	Low
GPU	Moderate	High	10's TFLOPS range	Low
DSP	Moderate	High	10's GFLOPS	Moderate to High
FPGA	Difficult	Moderate	Algorithm Specific	Moderate
ASIC	Very difficult	Low	Algorithm Specific	High

CPU, GPU, and DSP architectures are similar at the microarchitecture level. What differentiates them is how the microarchitectures are combined with each other and with memory and I/O.

FPGAs provide a semi-flexible solution where digital logic design is used to implement algorithms and I/O for a specific task. Modern FPGAs include a number of hardware multiply units that make them suitable for algorithms such as the FFT.

ASICs are custom chips that can achieve better performance than FPGAs. They are suitable for well defined algorithms.

EXAMPLE 1: THE SQUARE KILOMETER ARRAY

SKA TELESCOPE

SQUARE KILOMETRE ARRAY

Exploring the Universe with the world's largest radio telescope

The Square Kilometre Array (SKA) is a radio telescope with large antenna arrays located in multiple locations around the globe.

The name originated with the plan to have an effective capture surface of one square kilometer. The actual plan exceeds this original concept.

The SKA will be used to answer fundamental questions about the universe, such as the nature of dark energy, and probe farther into the universe to understand the early universe.

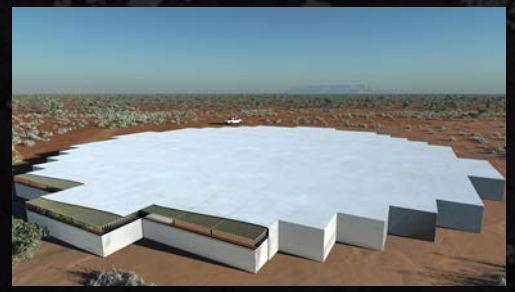
The raw data produced by the combined antenna arrays are 20x the size of current internet traffic and require supercomputers faster than what existed in 2015 to handle the processing of that data.

The construction of SKA will begin in 2018 and be ready for scientific use in 2020.

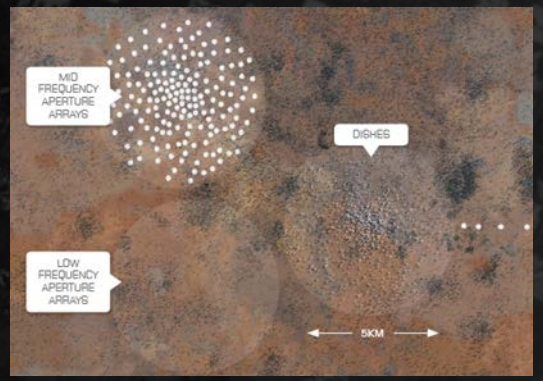
SKA DATA PROCESSING

Up to a million antennas per site

Form hundreds of arrays at each site



Multiple sites around the world



- Central location
- 100 + PetaFLOPS

Correlation

Beam forming

Data conditioning and transformation

Final Image Production

SKA ALGORITHMS

AMD

- Correlator (Cross Correlation):

  This is the Hermitian inner product that typically requires $O(N^2)$ operations. Array specific algorithms that require calibration can reduce this to O(N log N) operations. The correlator removes signal noise and combines the time delayed signals from each antenna.

- Beam forming:

  Beam forming is way of “steering” a fixed array towards a specific direction. This is done by combining the signals from individual antennas in the array with a time shift that aligns the phases of the signals that originate from a specific direction.

- Data conditioning and transformation:

  Removes corrupted data caused by interference or system faults and maps the data onto a rectangular grid. This is computationally expensive.

- Final image production:

  Utilizes FFTs and image processing techniques to produce a final image usable for scientific discovery.

EXAMPLE 2: WHOLE GENOME SEQUENCING

WHOLE GENOME SEQUENCING PROJECT

NIH will sequence 62,000 participants' genomes.

The goal is to establish genomic resources that represent the US population in terms of ethnicity and disease traits.

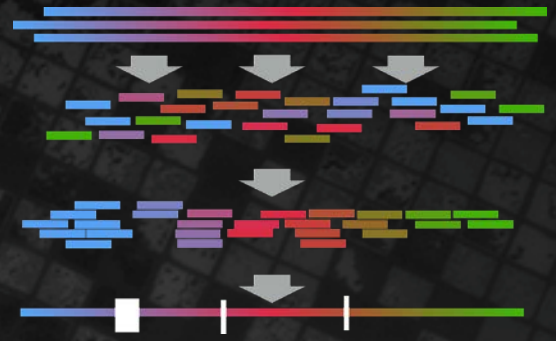
The intent of this project is to establish a genomic resource that represents the US population in terms of ethnicity, gender, and health.

The ability to efficiently sequence 62,000 individual genomes represents a challenge for computing in terms of data size and computational resources required.

WHOLE GENOME SEQUENCING (WGS)

Concept:

Assemble the entire genome from a large number of fragments. A genome is the complete DNA sequence for an individual organism.



Methodology:

The process starts with longer, but incomplete and overlaping, strands of DNA and fragments them into smaller pieces. These pieces are then assembled to form the complete DNA sequence using pattern matching techniques.

Data Size:

Raw data for 5000 genomes requires ≈180TB storage.

Processing Time:

Processing a block of 5000 genomes on a current system takes millions of CPU hours and 50 days of wall time.

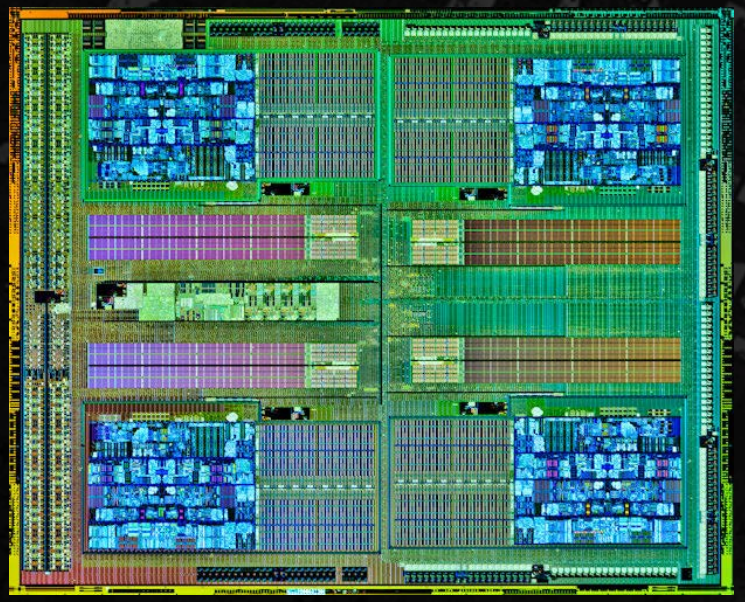
UNDER THE HOOD

TRENDS

AMD

- Heterogeneous computing and accelerators
- Increased on-chip integration. E.g. CPU + GPU on the same die
- Increased on-package integration using multiple chips on an interposer. Think: CPU + NIC + memory
- 3D or Die-stacking: Stacked memory chips and logic chips (as seen in recent GPU products such as AMD's Fiji)
- Higher core counts
- New memory technologies (e.g. NVRAM, stacked memory)
- Faster interconnects
- New programming paradigms: C++ AMP, OpenMP and OpenACC standards

CPU: SOME CURRENT TRENDS



AMD Zen CPU

CPU architecture trends:

- Bigger pipelines
- Increased out of order execution
- Improved speculative execution
- Wider vector operations
- Memory scatter/gather instructions (vectored I/O)

These architectural features improve performance at the cost of die space and reduced energy efficiency

Increasing core counts enables parallel thread execution

WHY WE OFTEN NEED ACCELERATORS

AMD

- Increasing the core counts results in increased die or package size for the CPU, increasing cost and energy utilization for all users regardless of need.
- CPU parallelism is limited to vector widths and core counts.
- Increasing the vector width requires the application programmer to develop algorithms that can take advantage of the wider vector widths. This is challenging for some science areas.

Accelerators address these problems by simplifying the core or by defining custom logic that implements specific algorithms.

GPU ARCHITECTURE DETAILS

AMD Hawaii GPU

Key Characteristics:

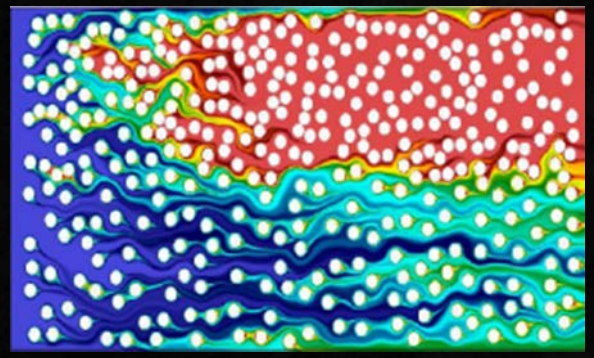
- Very high core count with highly parallel architectures optimized for parallel code
- Core architecture is simplified to reduce die space and improve energy efficiency
- Sequential code runs poorly on the GPU, although current GPUs have better support for general purpose compute
- Excellent floating point capability
- High throughput memory architecture

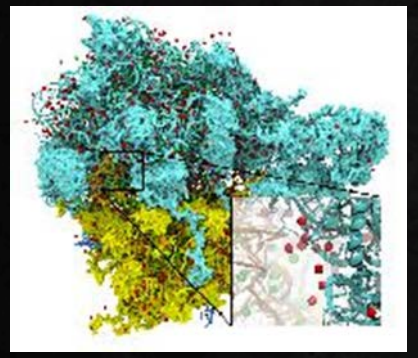
Programmable using OpenCL, C++ and other high level languages via OpenMP and OpenACC.

GPUs are good choices for highly parallel data processing such as the signal processing and image generation found in SKA as well as parts of the pattern matching algorithms found in WGS.

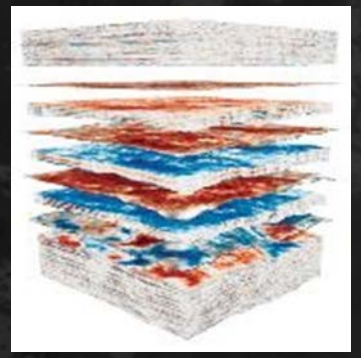
HANDLING MASSIVE DATA SETS AT MASSIVE SPEEDS

- A conventional CPU executes one thread at a time
- A multi-core CPU might execute tens of threads at a time
- A GPU can process *thousands* of threads concurrently
  - Repurpose pixel processing for general purpose processing
  - Huge increase in power-performance efficiency
- Highly parallel algorithms (e.g., X-correlation) experience massive acceleration
- Trend: accelerators are increasingly deployed to attack more algorithms and problems:


Computational Fluid Dynamics


Bioinformatics


Cosmology


Oil & Gas

FPGA ARCHITECTURE DETAILS

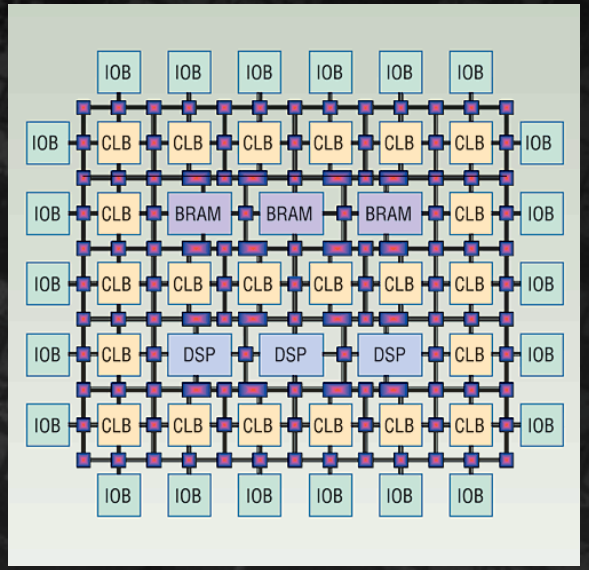
FPGAs are arrays of logic look up tables combined with signal routing fabric.

FPGA vendors often integrate small ARM CPU cores, DSP cores, memory controllers and common I/O IP into the FPGA fabric.

Programmable with Verilog. C/C++ programming now possible.

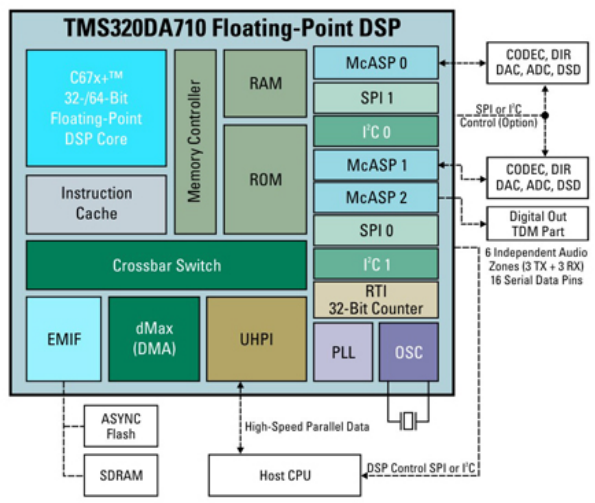
FPGA development requires an understanding of digital logic design principles.

Design tools help simplify large and complex chips that include integration of simple ARM CPUs, DSP CPUs, I/O and other IP blocks.



Source: IEEE Computer Society

DSP ARCHITECTURE DETAILS



Source: Texas Instruments

Key Characteristics:

- Highly specialized (limited applicability for general use)
- Small core count
- Limited floating point capability
- Optimized for high throughput digital signal data
- Tightly integrated with specialized I/O
- Low power
- Low cost
- Programmable using C and assembly

DSPs are well suited to signal processing algorithms such as digital filters and signal conditioning found in the SKA on-site processing.

COMPUTING HARDWARE CHALLENGES: $$$

AMD

Power Efficiency

- Geographically remote locations are required in SKA installations.
- WGS hardware is often located in facilities with power and cooling constraints

Cost

- Both SKA and NIH WGS projects are cost constrained. Commodity hardware is used in both projects to lower costs

SKA location: Murchison Region, Australia

(one of the most sparsely populated areas on Earth)

THERE'S NO "ONE SIZE FITS ALL"

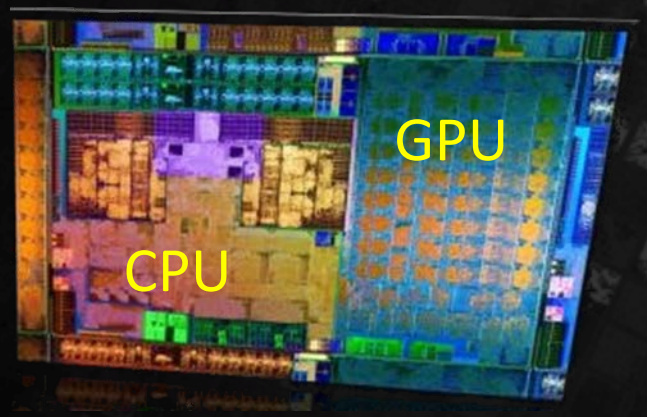
- Not all parts of a problem or algorithm are parallel and can be accelerated on parallel architectures
- High-performance CPUs are critical to address portions of codes that present few threads to the system

Not everyone wants to eat at the buffet

- Trend: increasing adoption of heterogeneous platforms with both CPU and accelerator (GPU) capabilities

CPU GPU

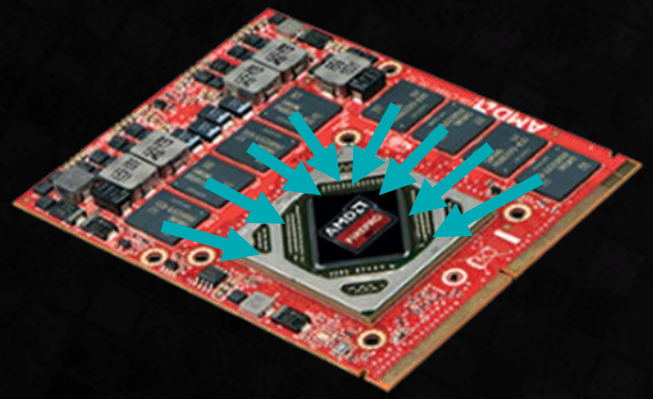
- Effective big data applications structure algorithms to make use of the best compute resources for each phase of computation



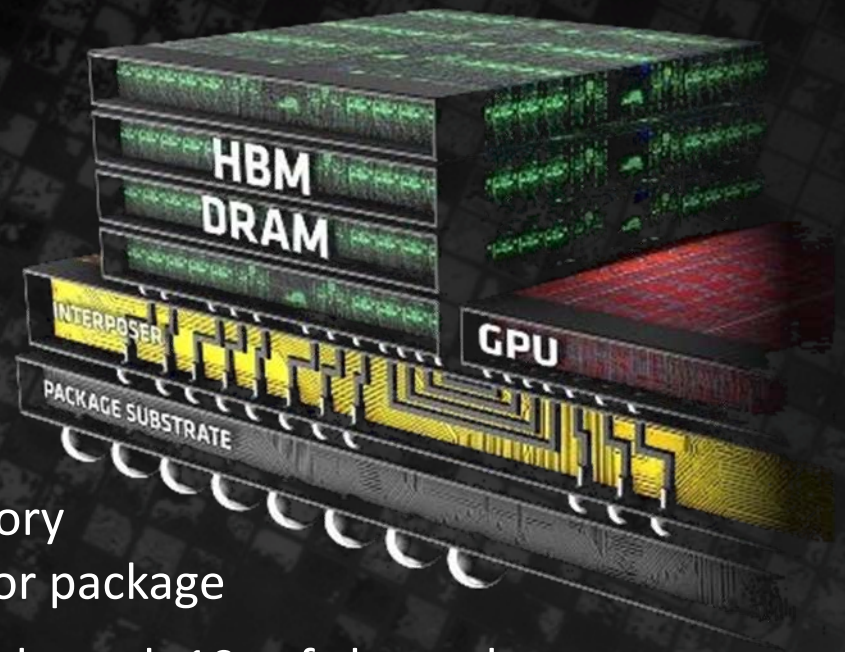
GETTING THE BIG DATA TO YOUR BIG COMPUTE

- With so much compute, data delivery becomes a critical bottleneck

Already a challenge for a few cores/threads



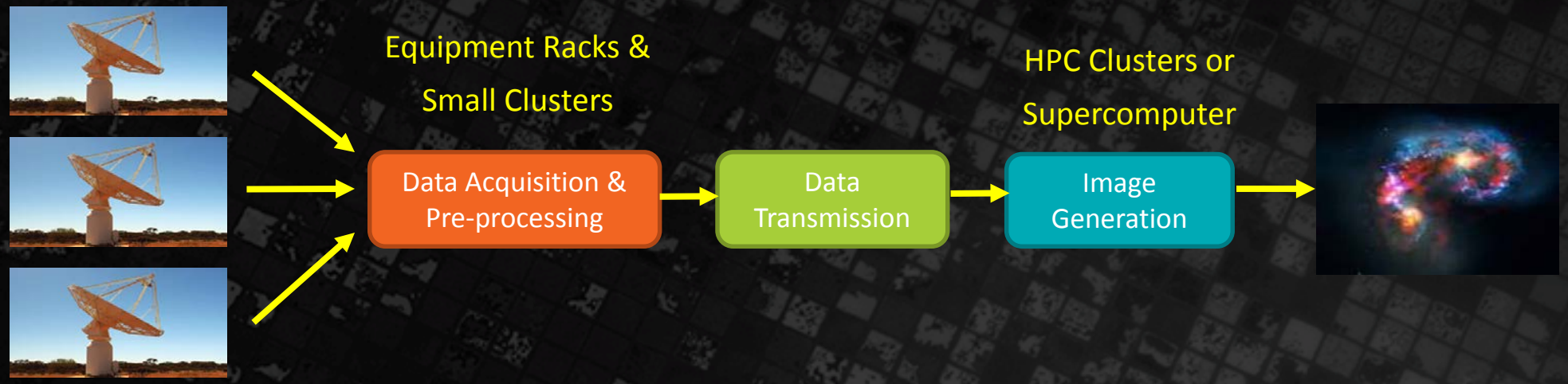
Traditional compute has only a few memory channels and 10s of GB/s of bandwidth



- Trend: integration of memory directly inside the processor package
- Provides TB/s bandwidths through 10s of channels

There are solutions to scale up bandwidth

SKA HARDWARE



- Antennas at each site are divided into groups for on-site, close to the antenna, data pre-processing. This reduces the data size required to be transmitted to the central location.
- Data pre-processing hardware combines ASICs, FPGAs and small HPC clusters with accelerators. This distributed processing approach makes possible processing of the combined petabytes/sec antenna data. ASICs and FPGAs are used to implement early pre-processing algorithms to achieve performance with the required energy efficiency. Sites use small HPC clusters to combine data prior to data transmission.
- The central location contains a supercomputer capable of more than 100 peta FLOPS required for the final processing that leads to images and data used in scientific discovery.

NIH WGS PROJECT HARDWARE

AMD

- To achieve the NIH goal of processing 62,000 genomes, a novel hybrid system can be used that combines cloud computing with local HPC clusters and large supercomputers.
- One recent study developed such a system capable of processing ~5000 genomes in 50 days. Amazon Web Services was found to be ideal for initial processing of the massive raw data. A supercomputer is used to handle the computationally intense portions of the data processing, but use policy limits jobs to 24 hour run times. Therefore, local HPC clusters are used for jobs that require > 24 hours to complete.
- The WGS project requires 12.5x the computing resources to achieve similar time to solution
  - 12 Amazon Web Services instances (may be possible)
  - 12 small clusters (possible)
  - 12 supercomputers (not likely to get simultaneous time on this many big machines)

The NIH could use an exascale-class supercomputer, HPC clusters, and enhanced cloud computing architectures that include the use of accelerators and advanced memory architectures.

Zhuoyi Huang, Navin Rustagi, Narayanan Veeraraghavan, Andrew Carroll, Richard Gibbs, Eric Boerwinkle, Manjunath Gorentla Venkata, Fuli Yu. **A hybrid computational strategy to address WGS variant analysis in >5000 samples**. *BMC Bioinformatics*, 2016; 17 (1) DOI: 10.1186/s12859-016-1211-6