High Performance Computing Technology Roadmap

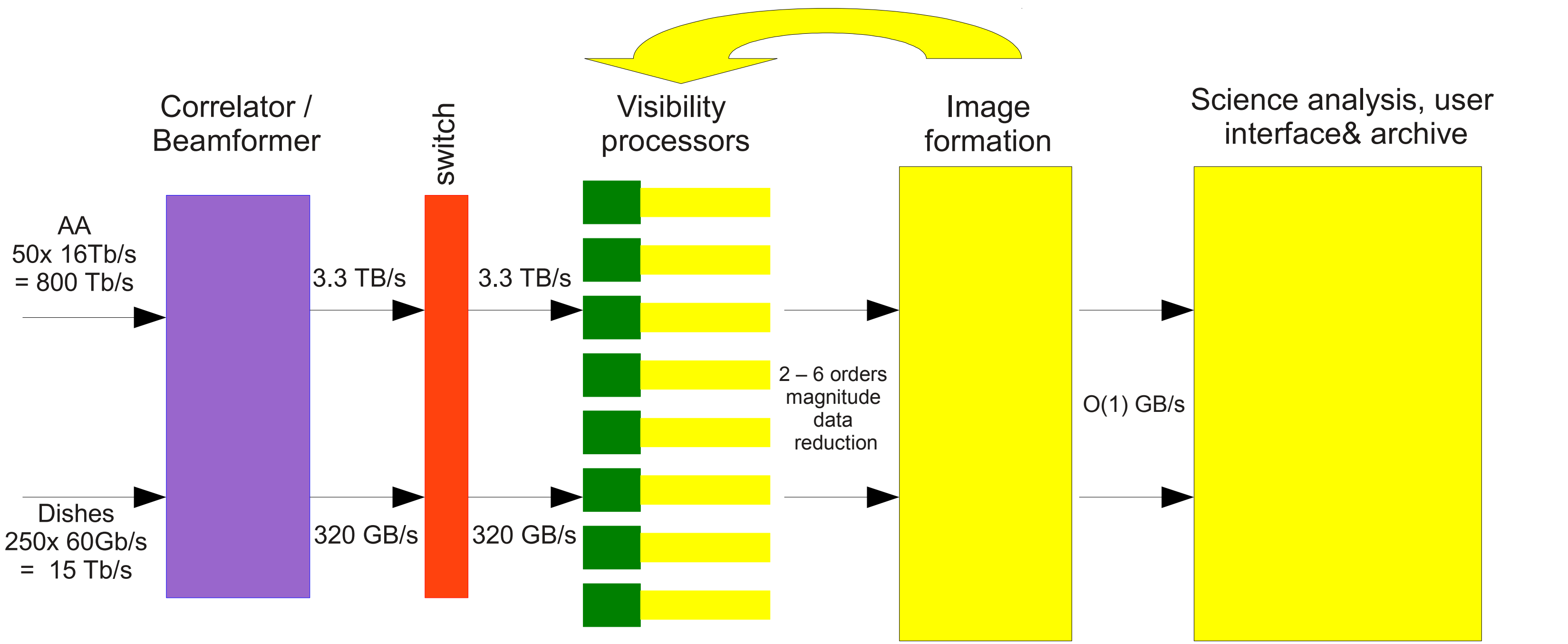
Background & context

- Preliminary SKA1 system description
  - SKA memo 130
- Well beyond current HPC systems
- SKA1 depends critically HPC
  - Improvements in HPC technology
  - Ability to efficiently use future HPC technology
- Identify risks and opportunities
- Necessary areas of research

Background & context

- Analysis of available public roadmaps
  - Exascale panel report (Kogge et al., 2008)
  - International Exascale Software Project Roadmap (Dongarra et al., 2011)
- Review current exascale initiatives
  - Lecarpentier et al., 2011
- Map to current SKA1 & 2 requirements

SKA1 Central Processor
(SKA Memo 130)



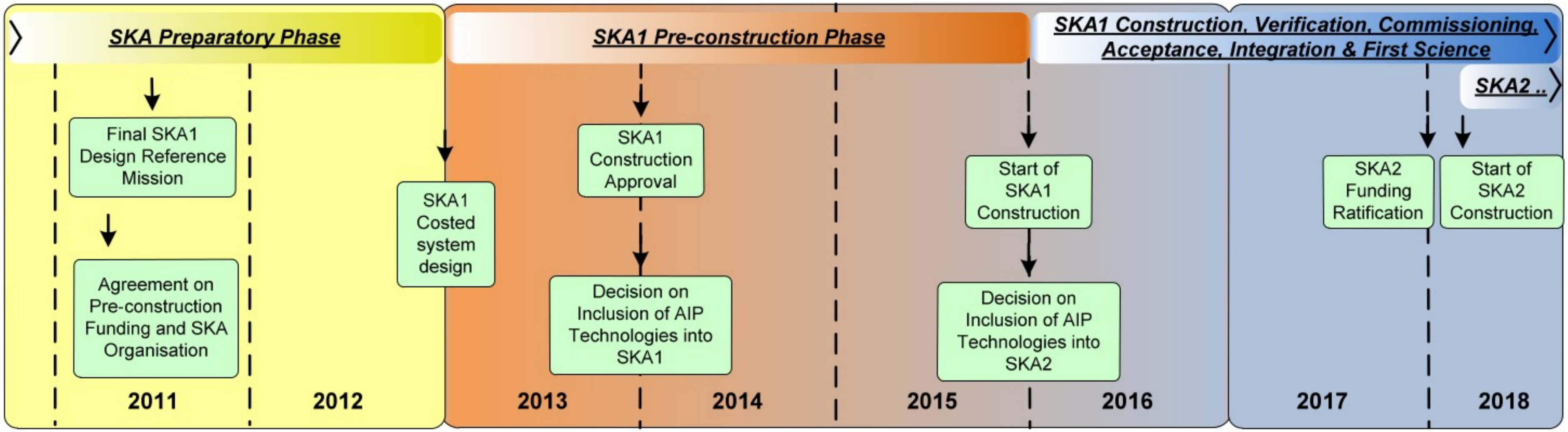
~ExaByte scale archive
Tiered structure
Archive sites at all major partners

<u>Minimum required Flops*</u>

AA: ~8 PFlops
Dishes: ~500 TFlops

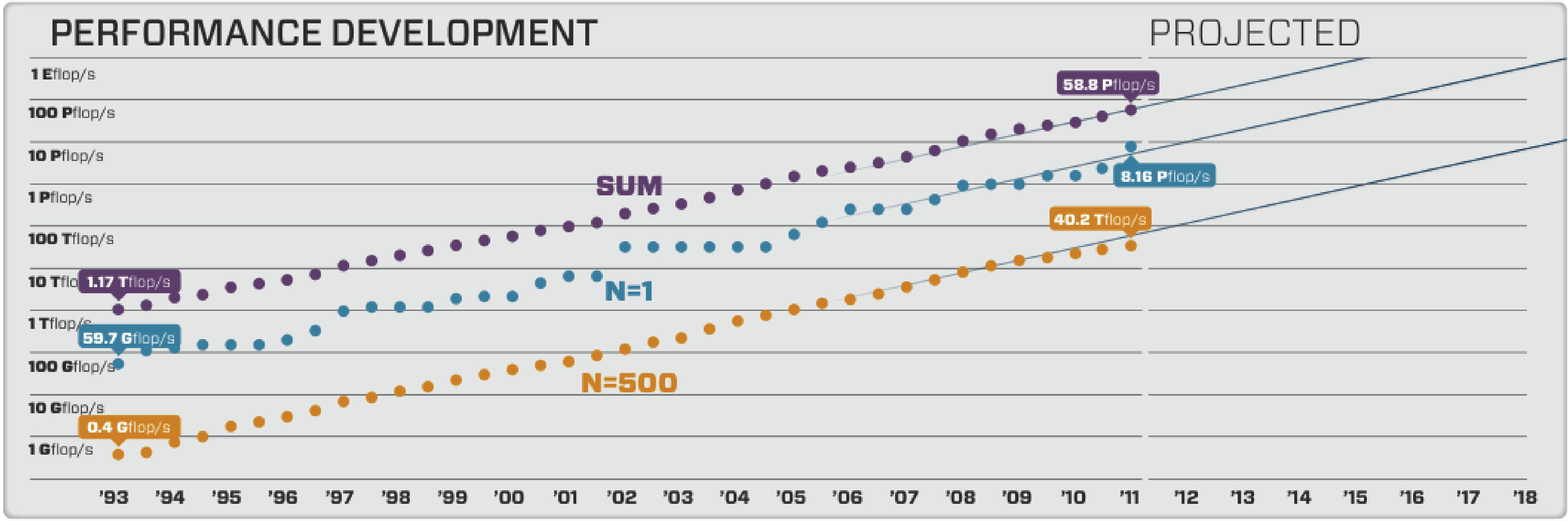
*Overall compute requirement may be a serious underestimate. This figure assumes 100% efficiency, and it is known that HPCs typically operate at much lower efficiency. Moreover, recent information indicates that 10^5 flops / float will be needed for wide-field, high dynamic range imaging. Thus the actual compute requirement may be more than an order of magnitude greater.

SKA top level phases and milestones



SKA1: ~2017
SKA2: ~2022

Projected performance of supercomputers



(source: top500.org)

SKA Compute Requirements

- SKA Phase 1 (to be procured ~2017)
  - Several hundred PetaFlop/s (SKA Memo 130)
  - Close to top of top 500
- SKA Phase 2 (~2022)
  - ExaFlop/s range
  - Again, close to top of top 500

SKA I/O Requirements

- External I/O often not part of roadmaps
- SKA Phase 1 output from correlator
  - 320 GB/s from dish correlator
  - 3 TB/s from Aperture array correlator
  - Challenging, but not a significant risk
- SKA Phase 1 input into correlator
  - Up to 10 - 500 TB/s
  - Without peer in the world, a significant risk

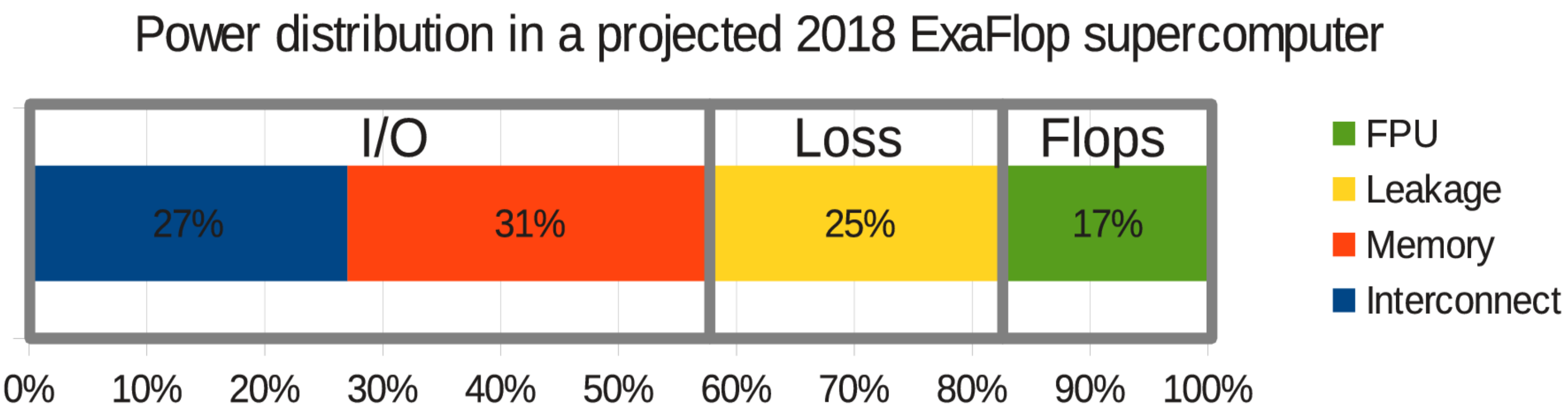
Projected development of HPC systems

Exascale Computing Study & top 500 lists

	2009[1] *Jaguar*	2011[1] *'K' Computer*	2018[2]	2009 vs. 2018
System peak	2 Pflop	10 Pflop	1 EFlop	*O(1000)*
Energy requirement	6 MW	9.9 MW	20 MW	*O(10)*
Energy/Flop	3 nJ/Flop	1 nJ/Flop	20 pJ/Flop	*-O(100)*
System memory	0.3 PB	1 PB	32 – 64 PB	*O(100)*
Memory/Flop	**0.6 B/Flop**	**0.1 B/Flop**	**0.03 B/Flop**	***-O(10)***
Node performance	125 Gflop	128 Gflop	1 – 15 Tflop	*O(10) – O(100)*
Memory bw/node	25 GB/s	64 GB/s	2 – 4 TB/s	*O(100)*
Memory bw/Flop	**0.2 B/s/Flop**	**0.5 B/s/Flop**	**0.002 B/s/Flop**	***-O(100)***

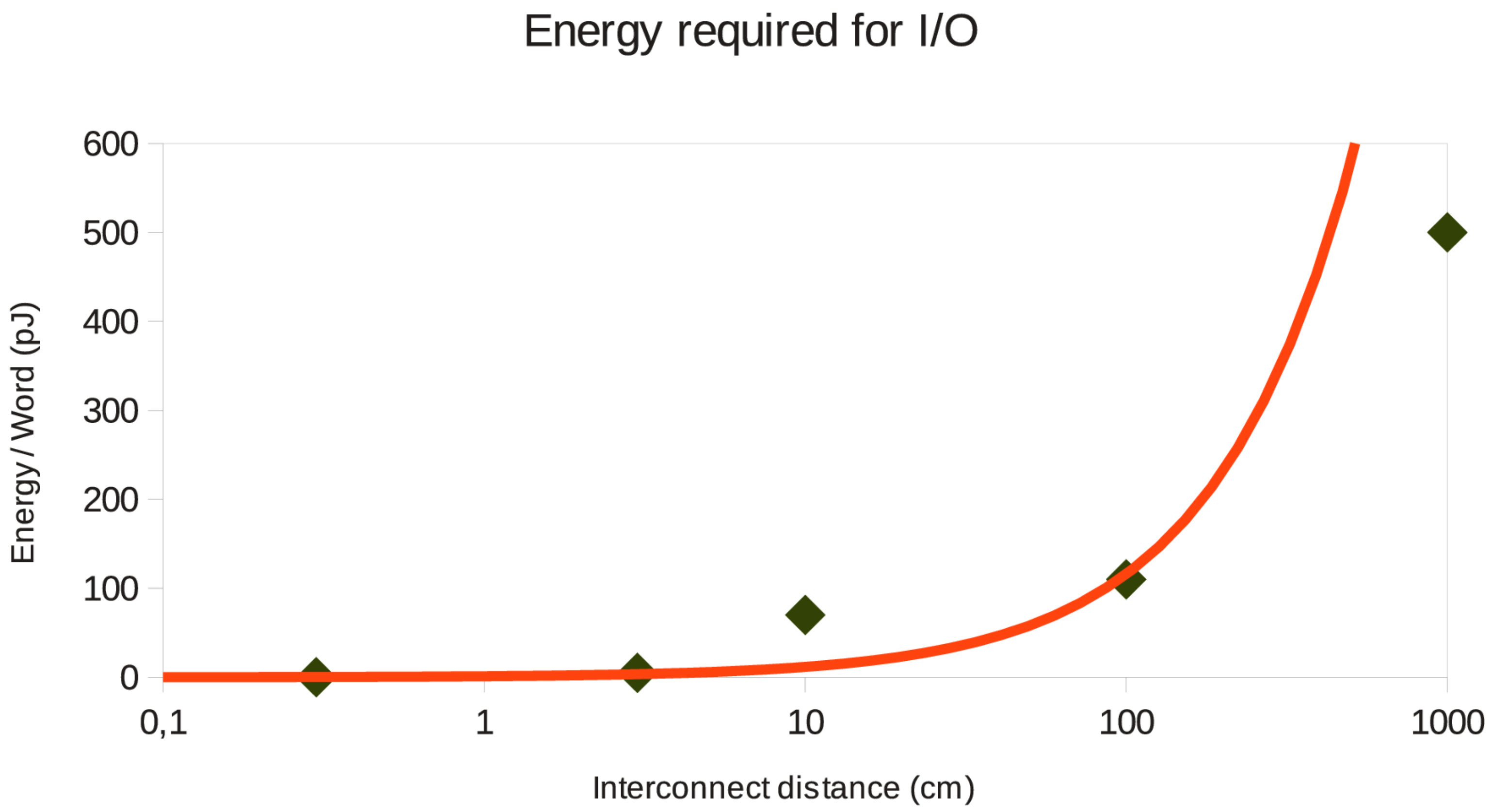
[1]www.top500.org
[2]ExaScale Computing Study, 2008

Power distribution in projected 2018 ExaScale 'aggressive strawman' system



(source: ExaScale Computing Study, 2008)

Energy required to transport a word of data



(source: Energy at Exaflops, Kogge, 2009)

The scaling challenge: the Walls

- Energy
- I/O
- Programmability
- Compute power
- Reliability

The scaling challenge: the Walls

- Energy
- I/O
- Programmability
- Compute power
- Reliability

- Energy dominates operational budget
- Needs revolutionary developments
- But all of industry is working on this
  - *Green computing*
- Optimizing code for minimal energy
  - Will save operational cost

The scaling challenge: the Walls

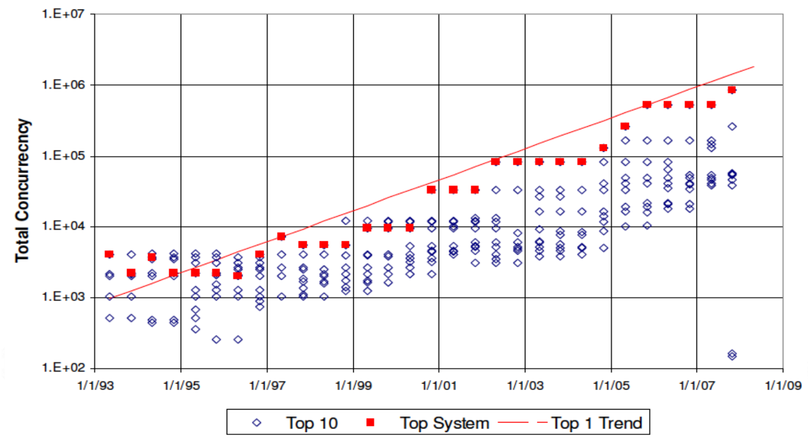
- Energy
- I/O
- Programmability
- Compute power
- Reliability

- Closely related to Energy
- I/O will/may dominate energy budget
- I/O will most likely be scarce
  - Memory bandwidth
  - External I/O
- SKA: low computational intensity
- Data intensive streaming processing
- Most effective way to reduce energy consumption: limit data movement

The scaling challenge: the Walls

- Energy
- I/O
- <u>Programmability</u>
- Compute power
- Reliability

- Hardware developments will introduce disruptive changes to software
- Massive parallelism
- Heterogeneous systems
- Many different levels of parallelism
  - Each programmed explicitly
  - Many different APIs
- Streaming processing, unique in HPC



The scaling challenge: the Walls

- Energy
- I/O
- Programmability
- Compute power
- Reliability

- Proven ExaScale problem
- Unique streaming data model
- Near real-time processing
- Low computational intensity
- Very high (input) data rates

The scaling challenge: the Walls

- Energy
- I/O
- Programmability
- Compute power
- Reliability

- Massive parallelism, massive number of components
- Components will inevitable fail
- System must be considered somewhat broken all the time
- SKA processing relatively robust against failures
- Frequency channels independent
- Failures can be handled, provided
  - Defects are detected
  - OS/software handles faults gracefully

Conclusions and recommendations

- Both SKA1 and SKA2 depend on HPC
  - Much work required to effectively use cutting edge HPC technology
  - Risk is high, mitigation steps to be taken
- Compute hardware probably available
  - I/O and memory bandwidth may be severely limited
  - But we don't really want a #1 top500 system
- Software may not
  - For data intensive computing
  - For real-time streaming processing

Conclusions and recommendations

- Aquire and maintain HPC expertise
  - Extreme scaling of data intensive applications
  - Real-time streaming processing
- Scaling gap: expand & strengthen collaborations with
  - Industry
  - HPC research community
    - (system research, not applications)
- Great interest in SKA from exascale community
  - Turn interest into useful research
  - 1st step: Kernels, compact applications, skeleton applications

Conclusions and recommendations

- Efficient use of exascale hardware requires complete redesign/rewrite of most software
- Depends on experts
  - Very small group of people
  - Intimately familiar with the processing required
- Lead with our HPC accomplishments
  - Notably LOFAR and ASKAP
  - 3 orders of maginitude scaling gap

Conclusions and recommendations

- Analyze critical reliance on HPC per use-case
  - Risk mitigation strategies for cases with substatial exposure
- Continue and expand partnerships
  - Industry, research institutes and exascale code owners
- Build and maintain expertise in extreme scaling of data intensive applications.
- Invest in the various international exascale projects
  - Focus on data intensive and near-realtime streaming applications
  - or systems particularly suited for these applications (hardware and software)
- Don't under-invest in algorithm development or HPC software expertise
  - Efficiency improvements may/will save operational costs
- Educate the SKA community about developments in exascale research
  - Add recurring exascale computing session to annual SKA workshop
- Internships at the various exascale software initiatives should be considered
  - Especially early in the pre-construction phase

Final words

- Disruptive changes are coming
- Most/all software needs redesign/rewrite
- Significant research effort required
- SKA has limited expertise in HPC
- We need outside involvement
- But: SKA generated great interest