

Panel:  
Software Challenges and  
Commonality

CHEP 2016  
Tuesday, 12 Oct, 2016

**Grid/cloud: interoperability,  
lesson learned, issues foreseen  
for the future or convergence -  
Frank Wuerthwein, Ian Bird**

- Predicting the future guided by fundamentals:
  - the “event” is an atomic unit of data, and we got lot’s of them  
**=> High Throughput Computing continues as dominant paradigm**
  - our ops budgets are dominated by salaries (physicists, engineers, technicians, students) instead of hardware  
**=> globally distributed computing continues as dominant paradigm**
  - significant fraction of computing is driven by conference deadlines  
**=> we will benefit from elastic scale out**
- **We will want a computing infrastructure that elastically scales out into commercial, HPC allocations, University IT not run by us, and other resources I can’t yet think off.**
- The elastic scale out ought to be controllable at all scales
  - from individual students, post-docs, faculty to institutions to national “grids” to centralized computing groups of the experiments to international (e.g. pan-european) purchasing organizations.

# POSITION STATEMENT - IAN

- NOT A QUESTION OF GRID VS CLOUD – BUT RATHER PRIVATE VS COMMERCIAL RESOURCES
  - WE NEED ULTRA-HETEROGENEITY IN ANY CASE
- TECHNOLOGIES ARE/HAVE CONVERGED:
  - GRID: FEDERATED, DISTRIBUTED; NOT RUN BY ME (RUN BY YOU)
  - CLOUD: DISTRIBUTED, (FEDERATED BY US); NOT RUN BY US (RUN BY SOMEONE WE PAY)
  - WE WILL ALWAYS HAVE A “GRID” IN THE STRICTEST SENSE (FEDERATION, RESOURCE SHARING)
- ISSUES:
  - WHAT IS THE USER/APPLICATION INTERFACE?
    - API OF EACH PROVIDER? OR HIGHER LEVEL E.G. HTCONDOR
    - SAME CONCERN FOR ALL HETEROGENEOUS RESOURCES (HPC, VOLUNTEER, OPPORTUNISTIC, ETC.)
    - ALSO FOR INTERACTIVE/ANALYSIS WORKFLOWS – NEED A USER INTERFACE THAT ALLOWS TRANSPARENT BACK-END SCALE OUT
  - COST: WHICH IS THE MOST COST-EFFECTIVE SOLUTION?
  - CAN WE GET ECONOMY OF SCALE – THROUGH JOINT PROCUREMENTS, OR BUILDING/FUNDING OUR OWN CENTRALIZED INFRASTRUCTURE?
  - RETAINING INDEPENDENCE: - DON'T WANT TO BE IN THE SITUATION OF OBJECTIVITY, ORACLE, ETC.
    - ALWAYS NEED TO BE ABLE TO SELECT APPROPRIATE RESOURCES TO SUPPLEMENT OUR OWN

**Code/Package/Software challenges:  
common packages & libraries pro/  
cons, technical or sociological barriers  
- Graeme Stewart, Liz Sexton-Kennedy**

# Graeme's Tao of Software...

- Change and challenge is continuous:
  - HL-LHC data rates and complexity, hardware, languages, old software, limited budgets, ...
- Software is a major, critical investment in HEP
  - Maintenance, upgrade, new ideas, ...
- Need to keep a vibrant HEP software community for the future
  - Cooperation is important... but it's not free
  - Nurturing and rewarding our coders... does writing good code advance your career?
- Community initiatives
  - HEP Software Foundation
    - Forum for organising the community steps beyond what we have done the past
  - Community White Paper
    - Present a forward vision — bazaar rather than cathedral, but let's be clear about what we want to build
- All of this must be geared towards *funding* people for software development across funding agencies
  - Did we learn the lessons of the past? Better to limit the scope than deliver baroque solutions late... maintainability loves simplicity...

# Software Commonality Position Statement : Liz

- Common software projects require trust in order to succeed. Examples from our past where we have succeeded:
  - Start with a product that is already successful for at least one organization
  - One or a few proponents then build trust with others and actively try to engage them so that they adopt and improve the software. Others trust that their needs will be addressed.
- Common projects also have to be sustainable
  - Simplicity and focus are important for the projects, but an openness to contributions is also important and successful projects balance the two
  - There is trust that the community will make sure that the product is supported for the length of time it is needed by it's stakeholders and users
- There must be a mechanism to set future directions and set priorities
  - Once a project becomes big enough and successful enough, some form of governance must be set up
- The open source model and tools developed to facilitate it, are an enabling technology that are field has started to capitalize on. We need to expand on this trend in order to increase the places where common solutions are technically possible.

**Infrastructure commonalities: HPC,  
HTC, exotic hardware, etc. What does it  
all mean for the communities and vision  
forward? - Wahid Bhimji, David Britton**

## Wahid's "Position"

- HEP will have a computing gap (60x++), HPC will have kick ass big machine (exascale et.al.)
  - HPC is keen on 'big data' - nersc and alcf call for data apps. DoE at least talks about 'hybrid' approach
  - There is no HEP workload these machines can't handle technically
    - 'Culture fit' is more of an issue - giant jobs; (relatively) frequent outages
- Exploiting these *will* require parallelization (in-node at least) and exploiting HBM (culture: 'Just strong scale to fit in HBM')
  - Such nodes may make into HTC too and we currently leave flops on floor of current machines from no vectorisation etc.
  - We (HEP) talk a lot about this - but 'it's just words, folks'
- (Experiment)HEP needs to be part of the HPC conversation
  - Vendors / CompSci / HPC Centres are keen for applications to work with / benchmark - (but who'd want to work with our code)
  - **Aim: most efficient solution we can rather than good enough**

- Funders are pushing for “joined-up” e-infrastructures to meet the needs of Big Science
  - They perceive economies of scale
  - Reduced costs of infrastructure development
  - Increased efficiencies (provision more for average, not peaks)
  - Advantages for development of useful skills and better career paths
  - Etc.
- But we know commonality is hard:
  - Historic lack of commonality between LHC experiments
  - Plethora of AA projects trying to make common access frameworks
  - Etc.
- We need to:
  - Provide Lego Blocks not complex systems.
  - Recognise that commonality does not imply one-size does not fit all - needs depend on size/expertise/use-case of the clients.
  - Take an “Economic” view: For example, investments in developing the ATLAS Event Service allows access to HPC, though one would never buy an HPC machine to run this workload.
- Challenges: AAA (technical, trust, policy); Building in flexibility to respond to future (unknown) paradigm shifts. Recognizing/evaluating “economic” future opportunities in time. Legacy code (parallelization!).

Flat-cash  
(at best)

Fall at the  
first fence?

Incremental and  
opportunistic