HPC in a Mist of Cloud Technologies and Infrastructures

Geoffrey Fox, Shantenu Jha

There are at least three trends that we see represented in any Future Platform for science research:

1) The increasing power and complexity of modern HPC systems as exemplified by those involved in drive to build exascale class machines.
2) The increasing use and sophistication of commercial and open cloud infrastructure (that can be used as IaaS, PaaS, SaaS, FaaS etc.)
3) The increasing functionality and use of Big Data systems in conjunction with HPC.

There are many other less critical trends and we quote a few

4) The growing use of Python and Jupyter notebooks as the user interface to both HPC and Cloud systems
5) The ability of large scale Simulations to produce Big Data outputs (visualizations) that require sophisticated analysis -- possibly in real time.
6) Scientific data is intrinsically distributed with multiple administrative domains (a key issue in Grid computing days)
7) The assertion that "any modern machine learning group must have deep learning and HPC expertise" which includes implicitly the observation that HPC is essential for good deep learning performance.
8) As well as virtualization approaches such as OpenStack, Docker (Singularity) is important with different performance-security trade-offs.

Although most publicity has been for commercial use of clouds, **we expect Universities and researchers to use both private and public clouds to a much great extent in the near future**. Current impediments include peripheral issues such as the financial model and network access, but both of these can be addressed. Even if it is only Universities that adopt clouds, it has impact on federal activities as these must be synergistic with academic research approaches.

In (3), we note importance of the Big Data systems associated with Apache Foundation, such as Hbase, Hadoop, Spark, Storm etc., which we term the Apache Big Data Stack (ABDS), even though important components such as MongoDB and Tensorflow are not Apache projects. We note that most of these technologies are in principle usable on both HPC and Cloud IaaS systems, though in practise many challenges remain. Independent of the hardware infrastructure, there are even stronger forces driving the adoption of ABDS technologies. They offer usability, functionality and sustainability that is not available in the HPC ecosystem.

We deduce that any Future Platform must satisfy the following constraints

a) It must allow easy integration of public and private clouds and allow HPC and cloud approaches to run well and run together

b) It must allow the powerful features of modern clouds such as ABDS, XaaS to be useable on HPC hardware
c) It must support distributed data sources and repositories
d) It should support modern workflow and portals including Python based front ends

We have found that gaining consensus as to way forward is handicapped by clear statements on requirements for science applications. Other confusions come from distinguishing needs of pleasing parallel applications from what can be called “global machine learning” with a single analysis treating large amounts of data distributed over the nodes of our Future Platform.

We use term cloud broadly without choosing particular implementations: public/private clouds, OpenStack/Docker virtualization. Probably public clouds either offering IaaS or those running today’s Internet are the lowest cost (but not necessarily lowest price) solution and in aggregate are far more powerful than the systems used in science research. Of course all systems require a significant ecosystem with many people developing, testing and running software. It is not clear to us how ecosystem costs vary between approaches and our interest in HPC-ABDS is partly aimed and reducing need to develop custom software and so increase fraction of ecosystem funding that goes to system hardware.

Case Studies: To demonstrate that traditional high-performance applications also require capabilities associated with clouds and how online supercomputing platforms **must** increase the variety and performance of analytic services without reducing traditional measures of performance and scalability we briefly analyze two application drivers:

As Exhibit A, we present precision medicine as a case study. Precision medicine such as the determination of optimal drug candidates tailored to suit genomic profiles suffers from the challenges of adequate sampling as well as the “curse of dimensionality”. Scalable simulations have to be coupled with smarter exploration of multidimensional phase space. This requires a coupling of extreme scale simulations capabilities [1] with a range of traditional analysis and new ML analysis [2].

As Exhibit B, we present the requirements of streaming applications, including those from experiments and observational systems. Streaming applications impose requirements of real-time processing and steering [3].

In order to support the requirements of Exhibits A and B, future science platforms must have a greater/richer set of analysis-as-a-service than currently available. Future analysis and associated middleware must utilize traditional performance capabilities, yet expose fundamentally new capabilities. Thus prudent, if not only approach, is to (re-)design the software stack for analysis. This requires the selective integration of the Apache Big-Data Stack (ABDS) capabilities appropriately implemented for supercomputing platforms. A realization of the HPC-ABDS concept is provided by the SPIDAL project [4] and discussed in publications [5].

1. CompBioMed: A Centre of Excellence in Computational Biomedicine http://compbiomed.edu
2. CANDLE https://cbiit.nci.nih.gov/ncip/hpc/candle
3. Streaming Systems: http://streamingsystems.org
4. SPIDAL Project http://spidal.org
5. http://hpc-abds.org/kaleidoscope/

===============

Increasingly, many traditional high-performance applications also require capabilities traditionally associated with clouds viz., strong co-location of compute, data and analytics services. Thus, future online supercomputing platforms **must** increase the variety and performance of analytic services without reducing traditional measures of performance and scalability. This requires a careful re-examination of the software stack, inter alia.

~~In this position paper, we use Medicine and~~

- ~~Exhibit A. Streaming requires hybrid capabilities (HPC + Clouds)~~
    - ~~Why do light sources use clouds? (if so).~~
        - ~~Embarrassingly Parallel ?~~
        - ~~Data costs/economics?~~

- ~~Exhibit B. Choose another domain -- such as medicine and show that more of the same (HPC) will not suffice.~~ ~~**For example: Medicine and Precision Analytics:**~~
    - ~~Data Access to large volumes of data.~~

- ~~Results of analytics stored in a data base.~~
- ~~Data-compute architecture.~~
    - ~~Latency tolerant.~~
    - ~~Blast~~
- ~~What is the HPC - medical school relationship at StonyBrook~~
    - ~~Mt Sinai, MSKCC, Broad Ins. etc ?~~

- ~~DOI/arxiv for streaming report.~~

CERES Streaming based on RFI

RFI 3 sections up to 1200 words each (https://www.nsf.gov/pubs/2017/nsf17031/nsf17031.jsp)
Research Challenges
CI needed
Other considerations e.g. training, sustainability

- Make explicit: 5 year timescale
- HPC-ABDS primary focus and streaming is a problem area that will benefit.
    - E.g. Personalized Medicine streams in but analysis isn't streamed.
- Technology Drivers towards HPC-ABDS:
    - Do clouds still have the same performance issues as they used to?
        - DL in Commercial Clouds.
    - Cloud security. Complex ML. Multitenancy issues in clouds?
    - Architecture for Pleasingly parallels + Extended Architectures

- Is there a compelling need for a Million core ML application, or a 100,000 servers running concurrently?
    - Single core vs modest clusters.
    - Unclear what is “modest”
- COMET → modest size (~1000 cores)
- Gateways… SaaS for Scientific Computing
- Platform Model:
    - (i) Data Center Model
    - (ii) Supercomputer
    - (iii) Comet Class Machine.
- Software Model:
    - Battery simulations (Classic SC)
    - PP (Multi-tenant OpenStack)
    - Modest Cluster Capacity (Docker)
- Architecture:
    - KNL
    - GPU FPGA
    - X86 continuation/ARM
- I/O
    - External bandwidth
    - SSD NVRam Classic …
    - HDFS Lustre ..
    - POSIX non POSIX (“everybody” agrees POSIX bad for scaling)
    - Communication needed
- How many distinct systems are there?
    - Commercially (i) (iii) called Cloud generalizing classic OpenStack clouds
    - If heterogenous as now commercially a system can support multiple platforms
    - Federal government prefers a multi-provider solution
- “Capacity Clouds” (CC) vs “Specialized SC”
    - The latter will not run small jobs.
    - 5 years ago, today and 5+: similar order, 1 order of magnitude differences, several orders
    - What is the user interface to Capacity Clouds?
        - SaaS
    - What is the developer interface to CC?
        - Units for services.
        - Add functions.
- What is the programming/development model?
    - GW are an interface.
    - *aaS --- agile approach cf Gateways.
    - community

Research Challenges

Simulations OK
Need data systems.
Long tail both but larger ratio data
Streaming applications are a particular challenge

Challenge is Convergence

Hardware support SC, capacity HPC, HTC (Hadoop), 16 bit DL, external I/O, internal disks and bandwidth, fog (edge), volunteer (mist)

Consume industry, economics based; whatever it is it will be called cloud
Can ask for virtual cluster with certain requirements e.g. 16bit, streaming bandwidth … e.g. software defined system is converged hardware. Provider (resource management software) gives you best possible virtual cluster. Small data centers will find it hard to provide efficient hardware

Software
refining/implementing/deploying HPC-ABDS is challenge

CI needed
Implement research challenge

Other considerations e.g. training, sustainability

Notes:
Hardware will not be universal. E.g. GPUs 16 bit vs 64 bit
Virtual technologies to instantiate.

Abstract* **Conceptualizing the Platform for Science beyond 2020**

A primary challenge to the OAC community is the need to define the Platforms for Science beyond 2020. We analyze major current trends and propose that in order to deliver the Platform for Science in 2020 the dominant research challenge is the convergence of capabilities of traditional HPC systems with richness of Apache Big Data systems. We call this the **cyberinfrastructure capability convergence (CCC)** challenge.

Research Challenge(s): *Describe current or emerging science or engineering research challenge(s), providing context in terms of recent research activities and standing questions in the field.*

There are at least three cyberinfrastructure trends that we see represented as we evolve the platforms for future science research:

1. The increasing power and complexity of modern HPC systems as exemplified by those involved in drive to build exascale class machines.
2. The increasing use and sophistication of commercial and open cloud infrastructure
3. The increasing functionality and use of Big Data systems in conjunction with HPC.

Point 1 is arguably self-evident and captured by march towards exascale systems.

For Point 2, although most publicity has been for commercial use of clouds, we expect Universities and researchers to use both private and public clouds to a much great extent in the near future. Future private cloud systems will support academic research. However, there are three orders of magnitude difference in the economics between commercial clouds and academic research. This means that commercial clouds will continue their evolution essentially independent and agnostic of academic research. Academia will adapt to use them.

In (3), we note importance of the Big Data systems associated with Apache Foundation, such as Hbase, Hadoop, Spark, Storm etc., which we term the Apache Big Data Stack (ABDS), even though important components such as MongoDB and Tensorflow are not Apache projects. We note that most of these technologies are in principle usable on both HPC and Cloud IaaS systems, though in practise many challenges remain. Independent of the hardware infrastructure, there are even stronger forces driving the adoption of ABDS technologies. They offer usability, functionality and sustainability that is not available in the HPC ecosystem.

The research challenge is to provide software systems that support the richness of the ABDS stack but have the performance of traditional HPC systems. We call this research challenge the **Cyberinfastructure Capability Convergence** challenge and believe will shape if not dominate CI research in different ways.

A platform that supports CCC must be usable by applications from both ends of the spectrum: traditional HPC applications that need Big Data ("All Exascale Applications are Big Data Problems"), as well as big data applications that will need HPC more and more (e.g., Deep Learning with HPC capabilities).

Cyberinfrastructure Needed to Address the Research Challenge(s)

In order to achieve **CCC** future Platform for Science will need research that integrates across three dimensions: Software and Architecture. The options along these will be:

- **Macroscopic Architecture:** The three primary macroscopic architectures are: (i) Data Center Model, (ii) Traditional supercomputers and, (iii) Clusters (with virtualization) such as those represented by Comet.
- **Microscope Architecture:** The three primary microscopic architecture are: (i) Continuation of X86 systems, (ii) Many core systems (e.g., KNL) and (iii) non-traditional architectures (e.g., GPU, FPGA) etc.
- **Application Software:** Independent of Architecture, there will be a need to support CCC across (i) classic MPI-based simulations, (ii) pleasingly parallel and workflow systems, and (iii) data-intensive applications epitomized by deep learning.

A platform that supports CCC will be a software-defined system that works across different types of macroscopic and microscopic architectures as well as for different applications systems. Given divergence across above three dimensions, it is possible that multiple platforms (i.e. software-defined system) that support CCC emerge.

Question 3 Other considerations (maximum ~1200 words, optional): *Any other relevant aspects, such as organization, process, learning and workforce development, access, and sustainability, that need to be addressed; or any other issues that NSF should consider.*