Non-Traditional Workloads @ RACF

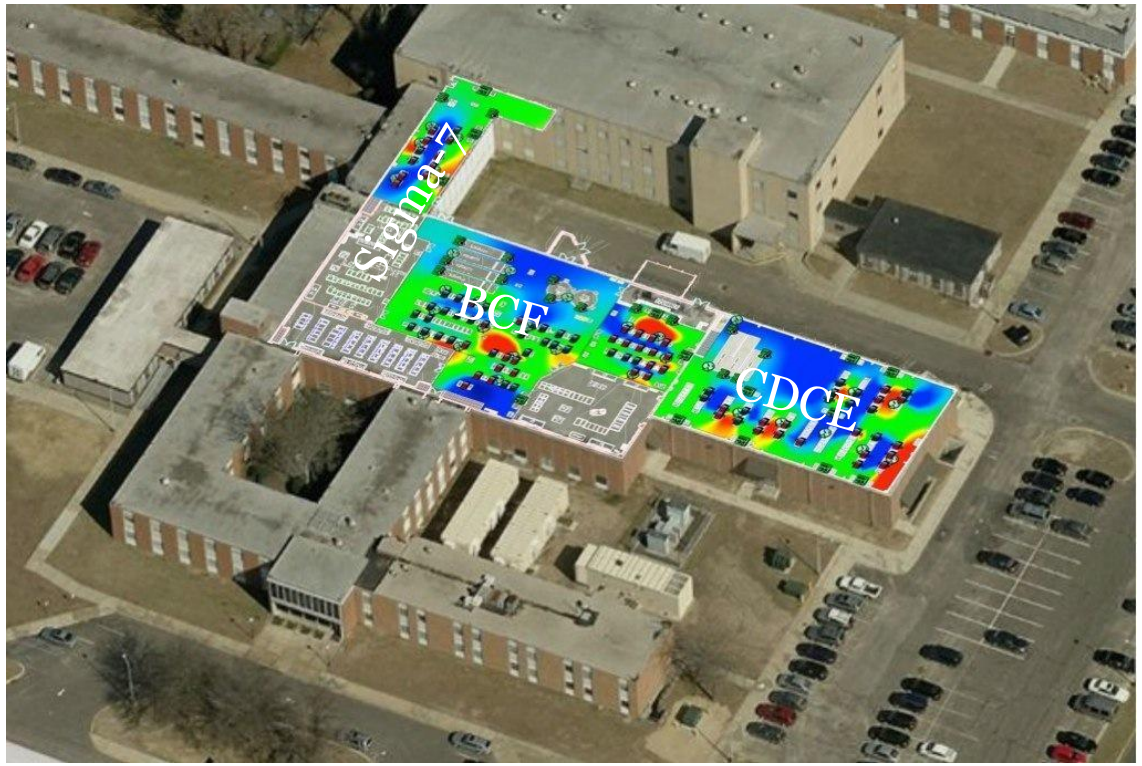
Running non-HEP/NP and non-HTC workloads in a traditional HTC environment

Tony Wong, William Strecker-Kellogg

HEPiX Fall 2015

History

- RHIC / ATLAS Computing Facility
  - Two experiments with a very traditional workload
    - Scale is the challenge, not computational framework
  - Expertise in “big data” and HTC
- Similar to other HEP/NP facilities
  - Growth in computing in physics organic and gradual, all things considered
- Little to no experience with HPC technologies



New Science needing Big Computing

- New detectors at light sources, electron microscopes, and others are suddenly producing data orders of magnitude faster than before
  - No institutional history of large-scale computing
- Not to put too fine a point on it, HEP/NP has been here for decades
  - Others just becoming familiar with concepts of storage arrays, batch systems, etc…
- Previously may have used one desktop with a GPU for satisfying all computing needs

New Users

- National Synchrotron LightSource II
  - Other similar lightsources worldwide
- Center for Functional Nanomaterials
- Data volumes from detectors orders of magnitude higher
  - Shigeki's site report detailed some of the networking challenges
- New paradigms for processing the data
  - Scaling up from one-desktop-with-a-gpu to entire compute cluster
    - HPC? HTC? something in-between?
- Lack of experience with "traditional Big Data"

RACF as Microcosm of HEP/NP Compute

- Many, many, facilities out there of roughly the same scale and capabilities as the RACF in HEP/NP
  - Embarrassingly parallel workloads, HTC oriented
  - Data storage as large a concern (or larger) as compute
  - Batch relatively simple—provisioning/matchmaking, not scheduling
  - Well staffed groups with large institutional knowledge of computing
    - New users can take advantage of documentation and expertise
    - Dedicated experiment liaison positions for helping onboarding new users

New Paradigm

- Users of these facilities can change on a weekly basis
  - No traditional “experiment” as a repository of computing knowledge
  - Each new beamline user is *really* new
- Software situation is chaotic at best
  - Various poorly supported open source or free/abandon-ware
  - Some commercial offerings, not suitable for use in a cluster facility
  - Interactive / GUI software integration into workflows
    - Sometimes exclusively interactive, or requiring “fast feedback”

Communication Challenges

- New users may not even be aware of their own computing needs
- Challenge of communicating across fields
  - Managing expectations
  - Different spheres of technical expertise
  - Conflicting terminology
- Multiple “experiments” being done on one apparatus, no shared place for communicating computing requirements / experiences

Sharing Knowledge

- No question that new computing paradigms will be needed
  - Scope of science and diversity of computing requirements

- The $10,000 question:

  Can we leverage existing infrastructure?

  - During the transition, or can it fulfill a permanent need?

HTC/HPC Divide

- Is this a false dichotomy? Or at least a blurry line?
  - Not for the obvious cases—LHC vs Fluid Dynamics
  - There could be a middle ground
- When scaling up from desktops, how much HPC is needed?
- Jobs needing 8 nodes 6 years ago can now fit on one 32-core node
  - How much of new workloads can be accommodated without going “full HPC”
    - InfiniBand / Slurm / MPI / etc…
  - Benefit is leveraging existing HTC facilities like RACF or even DHTC like OSG
  - What would a hybrid look like
    - Can HTC / HPC scheduling coexist?

Running at RACF/HTC Facilities

Zero-Order Requirements

- Embarrassingly Parallel
  - (small) Input → (one) Process → (small) Output
  - No communication of intermediate results
- X86_64
  - Other hardware not standard in the community
- Data accessible
  - May seem obvious, but need adequate bandwith to get data to the compute and back
    - Something to think about of moving from single desktop

Running at RACF/HTC Facilities

First-Order Requirements

- Linux (RedHat)
  - Virtualization is an extra complexity, Windows expensive
  - Containers / Docker allows simple cross-linux compatibility
- Free Software
  - Instance-limited licenses are hard to control across many pools
  - Cost of licenses becomes prohibitive with exponential computing growth
- “Friendly” resource profile
  - Code runs not just within the machine, but within the general limits its neighboring jobs use

Case Study

- CFN user was able to write their scripts in python, using numpy, scipy, and PIL, on a Windows desktop
  - Hopefully not an outlier as far as workflows go
- User was not experienced with Linux
- In the end, they ran their workflow successfully

- Recently helping biology with a small, temporary HTCondor parallel-universe MPI deployment

- Facilitated them to run in spare cycles on the RHIC compute farm
  - Concepts
    1. Log in and shell
    2. Global filesystem (GPFS)
    3. Software packaging and dependencies
    4. Batch systems, how to parameterize by input file

Liaison Position

- Ideally: a computing liaison, provided by the interested user-groups
- Someone to facilitate communication between users and facilities
  - Understand enough scientific computing & infrastructure to know what technologies are useful to even look at
    - "This needs MPI" or "GPUs could help here"
- Facility facilitator for the rapidly shifting user-groups
  - Becomes more a management / communication problem
  - Really need a ready-to-use solution that fits most cases

Liaison Position

- Avoid a “tower of babel” from joining two worlds with overlapping nomenclature and differing understandings / expectations
- Liaison position useful for avoiding unproductive novice-user ↔ admin communication
  - Basic training
    - Shared filesystems “you mean my data is already there?”
    - Batch systems “I can run more than one of my processes?”
    - Managing expectations “I may have to wait in line?”

Conclusions

- New facilities are popping up that require new computing paradigms
- Perhaps we can leverage our experience in HEP/NP to aid
  - Know fit won't be perfect, but something is better than nothing
- Software availability is larger problem—ignored for now
- Rapidly changing userbase needs to be brought up to speed and given a working solution quickly

How have other facilities tackled these problems?

The End

Thank you for your time
Questions? Comments?
willsk@bnl.gov