Diversity in Computing Technologies and Strategies for Dynamic Resource Allocation

Gabriele Garzoglio, Oliver Gutsche
21\textsuperscript{st} International Conference on Computing in High Energy and Nuclear Physics (CHEP2015)
15. April 2015
What keeps us up at night! (I)

- The nature of science:
  - Probe nature in more and more detail
  - Instruments get more powerful
  - Processing and analyzing data gets more resource intensive

- Software development is making continuously big strides
  - Improving software efficiency significantly

- But increased complexity of data is driving ever increasing resource demand

- Question: How can we provide access to sufficient resources? \(\rightarrow\) CAPACITY
What keeps us up at night! (II)

- Activity of experiments is not constant
  - It varies significantly with external triggers
    - Operation schedule
    - Conference schedule
    - Holidays, vacation time, etc.

- Question: How can we provision resources efficiently? ➔ ELASTICITY
This talk

- In the recent past, HEP resources were firmly based on **Grid** technologies
  - HEP applications == HTC
    - High Throughput Computing applications

- The need for more **capacity** and **elasticity** makes us look at other resource providers:
  - Cloud
  - **HPC** = High Performance Computing
10,000 feet overview

**Grid**

- Virtual Organizations (VOs) of users trusted by Grid sites

- VOs get allocations ➔ Pledges
  - Unused allocations: opportunistic resources

**Trust Federation**
10,000 feet overview

**Grid**
- Virtual Organizations (VOs) of users trusted by Grid sites
- VOs get allocations → **Pledges**
  - Unused allocations: opportunistic resources

**Cloud**
- Community Clouds - Similar trust federation to Grids
- Commercial Clouds - **Pay-As-You-Go** model
  - Strongly accounted
  - Near-infinite capacity → **Elasticity**
  - Spot price market

**Trust Federation**

**Economic Model**
10,000 feet overview

**Grid**
- Virtual Organizations (VOs) of users trusted by Grid sites
- VOs get allocations → Pledges
  - Unused allocations: opportunistic resources

**Cloud**
- Community Clouds - Similar trust federation to Grids
- Commercial Clouds - Pay-As-You-Go model
  - Strongly accounted
  - Near-infinite capacity → **Elasticity**
  - Spot price market

**HPC**
- Researchers granted access to HPC installations
- Peer review committees award **Allocations**
  - Awards model designed for individual PIs rather than large collaborations
Challenges

- How can we use Clouds and HPC installations with HEP HTC applications?

- How can we transparently integrate them into our Grid-based setups?

- How can we marry the different allocation models (static, economic, grant)?
Grid
The Grid

- The Grid is many things
  - Allows transparent access to a vast amount of resources
  - Solved the authentication problem
  - Established a trust model

- Federation of resources

- The Grid is successful
  - The LHC experiments were amongst the first to rely on the Grid ➔ Worldwide LHC Computing Grid (WLCG)
  - National Grids are successful to bring large scale computing to “smaller” science communities
Grid submission today: Pull Era of the Grid

- Submission evolved from earlier days of the GRID ➔ pilot-based submission infrastructures

  - Create overlay batch system ➔ Essentially a virtual pool spanning multiple sites
    - Use lightweight pilots to claim resource
    - Pilot pulls work and executes it

- Advantages
  - Late binding ➔ control scheduling on global scale
  - Reduction of failure rate ➔ job execution only starts after resource was successfully claimed
  - Integration of new kinds of resources ➔ simply enable resource to run pilot and pull work

- Disadvantages
  - Debugging problems is more complex

- Example: glideinWMS
  - What I like about the concepts implemented in glideinWMS:
    - Provisioning systems can be shared amongst many different communities
    - Separate pools of resources can be provisioned per community
    - Community has control over policies/priorities within its pool
HPC
HTC on HPC installations

▪ HTC: High Throughput Computing
  * Independent, sequential jobs that can be individually scheduled on many different computing resources across multiple administrative boundaries(*)

▪ HPC: High Performance Computing
  * Tightly coupled parallel jobs, must execute within a particular site with low-latency interconnects(*)

▪ Long history in HEP in using HPC installations
  * Lattice QCD and Accelerator Modeling exploit the low latency interconnects successfully for a long time

▪ Traditional HEP framework applications are starting to get allocations awarded
  * unmodified if HPC is intel-based
  * cross-compiled if HPC is non-Intel-based

▪ In all cases, allocation is proposal-driven and awarded through peer review committees
  * Examples are separate committees for all HPC installations funded by either the National Science Foundation (NSF) or the Department of Energy (DOE); or by committees specific to individual installations
  * Differ in proposal requirements from demonstrating technical capabilities to relevance of scientific research

San Diego Supercomputer Center (SDSC)

- Example for intel-based HPC installation
  - SDSC operates wide range of HPC clusters ranging from ~10k to ~50k cores

- Allocation award procedure
  - Individual Principal Investigators (PIs) submit proposals
  - Committee meets every 3 months to award allocations
  - Successful proposals have one year to use their allocations
    - Follow-up proposals need to demonstrate scientific impact

- CMS was awarded first grant in 2013 to re-process specific primary datasets (HTMHT & VBF)
  - Used pilots submitted through ssh login node
  - Follow up grants and proposals are progressing well, other experiments are equally successful
  - New: CE access to SDSC clusters simplifying access
Mira at Argonne National Laboratory

- Example for non-Intel-based HPC installation
  - Mira (PowerPC, ~49k nodes each 16 cores, almost 800k cores)

- Similar Allocation Award procedure
  - Proposals need to demonstrate enabling new science through using Mira

- Generating Atlas LHC Events on Argonne
  - Necessary: Alpgen (Fortran-based HEP event generator) recompiled using IBM XL compilers
    - (Effectively using MPI to run N-instances of Alpgen in parallel)
  - Mira has minimum partition size (512 nodes)
    - Opens ability to effectively use ‘backfill’ queues which can yield ‘free’ computing time.
    - Jobs are submitted by a custom workflow system with the goal of integrating Mira into the Atlas production workflow system.

For more information, see Taylor Childers Track 8 parallel session contribution on Thursday afternoon: “Simulation of LHC events on a million threads”
Cloud
Cloud Allocation Model - The peaks...

- The activity of the experiments is not constant!
  - It varies significantly with external triggers
    - Instrument operation schedule
    - Conference schedule
    - Holiday festivities, vacation time, etc.

- There might be a solution on the horizon:
  - Commercial Clouds

- Commercial Clouds offer “Pay-As-You-Go”
  - Offer scaling to infinite (...very large...) capacity on short time scales ➔ Elasticity
    - “There is no difference in price at AWS when using 1 CPU for 1000 hours or 1000 CPUs for 1 hour” (Jamie Kinney, Sr. Manager, AWS Scientific Computing)

- Can we use commercial clouds competitively to fulfill our peak demands?
Cloud Allocation Model - ... and the solution?

- It’s all about scale!
  - What are the challenges we face to run at scale on commercial clouds?
  - As an example: concentrate on Amazon Web Services (AWS)

- Many HEP experiments and facilities are working with AWS

- Goal: Improve integration with HEP workflows
  - Examples: Atlas, CMS, STAR, NOvA*, etc. / BNL**, FNAL, etc.
  - It’s all about understanding how most efficiently to use AWS capabilities

- Several areas of work
  - Provisioning
  - Economic model
  - Networking
  - Storage
  - On-demand Services

---

* Parag Mhashlikar: "Cloud services for the Fermilab scientific stakeholders", parallel talk in track 7 on Thursday

* Andrew Norman: "Large Scale Monte Carlo Simulation of neutrino interactions using the Open Science Grid and Commercial Clouds", poster session A

** John Hover at HEPiX Spring 2015: "Running ATLAS at scale on Amazon"
Provisioning

- Provisioning straight forward
  - Use cloud interfaces to include resources in pilot-based submission infrastructures

- We can provision resources on AWS by …
  - … paying for regular instances and design the instances to our needs
    - Our instances need: enough memory and local disk for our jobs and ability to run long enough to complete our jobs
  - … by using the spot price market
    - AWS spot pricing: bid your top price and pay the market price until it goes above your bid → disadvantage, instance might go away when market changes
    - Works well as long as spot price is stable on time scales >> than typical runtimes and/or workflows can deal with pre-emptable Grid cycles

- Integration challenges:
  - Develop mechanisms to expand and contract provisioned resources while job queue is full
  - Important to provision sufficient resource on the spot price market: integration with AWS availability zone management
Economic model

- Cost of 1 CPU h at AWS compared to our facility costs (order of magnitude):
  - AWS 1 vCPU regular instance (m3.medium) per core ➞ ~ $0.07
  - BNL 2013 estimate at RACF* per core ➞ ~ $0.04
  - Fermilab 2011 estimate at FermiCloud per core ➞ ~ $0.03
  - AWS 1 vCPU spot pricing (m3.medium) per core ➞ ~ $0.01

- To exploit elasticity need detailed understanding of cost model
  - Benchmarks of our workflows very important
  - Detailed understanding of characteristics of our workflows helps optimizing costs
    - Example: HEP applications can deal with arbitrary number of cores if memory and local disk is large enough ➔ industry prefers resources with fewer cores

- Integration challenges:
  - Reliable comparison of provider’s unit computation core (e.g. AWS ECU) and “standard” Grid equivalent (e.g. HS06)
  - Determine metrics for cost model, for example: I/O characteristics, service needs, data volumes, etc.

---

T. Wong at HEPiX Fall 2013: "Operating Dedicated Data Centers - Is it cost-effective?"
Scale test example

- Test to run at scale on both owned and rented resources
  - Results obtained through a 3-years collaboration with the Korean Institute of Science and Technology Information (KISTI)

- Up to 1000 jobs run simultaneously on each AWS and FermiCloud at Fermilab
  - Compute charges $398 ($0.14 per machine/hr), 525 VM’s
  - $51 of data transfer charges.

- Lessons learned
  - Commercial clouds charge for outgoing data transfers!
    - Needed to optimize jobs to reduce data transfer charges
  - Jobs need services to run!
    - Naive model using services provided externally has its limit
    - First trial overloaded the CVMFS stratum 1 infrastructure at Fermilab

NOvA use case:
- Parag Mhashlikar: "Cloud services for the Fermilab scientific stakeholders", parallel talk in track 7 on Thursday
- Andrew Norman: “Large Scale Monte Carlo Simulation of neutrino interactions using the Open Science Grid and Commercial Clouds”, poster session A
Storage

- Optimization of storage interaction of our workflows is crucial
  - Outgoing network bandwidth capacity is limited and needs to be payed for

- 2 main strategies for data transfers
  - Fill the available network transfer by having some jobs wait → Pay for idle resources
  - Store data inside cloud (AWS: S3) and transfer data back asynchronously → Pay for data in S3

- R&D will be necessary to optimize storage interaction
  - The cheapest strategy depends on the storage bandwidth, number of jobs, etc.
Networking

- Attack data transfer costs from a different angle

- Implement peering of our scientific networks directly with AWS infrastructure
  - Utilize upfront investments in scientific networks
  - Example: ESNet peering with AWS availability zones in the US

- AWS / ESNet data egress cost waiver
  - Transfer charges are waived for data costs up to 15% of the total bill - if network transfer goes exclusively through ESNet

John Hover at HEPiX Spring 2015: “Running ATLAS at scale on Amazon”
On-demand Services

- Jobs depend on services to run, they can be deployed
  - at sites outside the clouds
  - inside the cloud

- In both cases, they have to be dimensioned correctly to scale sufficiently
  - Services include data caching (e.g. Squid) WMS, submission service, data transfer, software delivery (e.g. CVMFS stratum 1), etc.

- Automating the deployment of these services on-demand in clouds enables scalability and cost savings ➔ active area of R&D
  - Use classical scaling techniques using service discovery, central name service, centrally controlled additions/removal
  - Clouds provide their own orchestration layers (for example AWS CloudFormation) which take care of on-demand scaling even more efficiently
Next Step: Educational Grants from Amazon

- Amazon currently works with different experiments/institutes to bring HEP use of Cloud resources to reliable, production use
  - Atlas is currently leading this area, CMS and Intensity Frontier are utilizing own grants

- Example grant for Intensity Frontier experiments at Fermilab:
  - Run data-intensive NOvA applications and Neutrino Beam Simulations on AWS
  - Considering adding other use cases from Intensity frontier experiments

- To put this test into context
  - FermiGrid (without CMS resources) has capacity of 145 million hours/year
    - NOvA alone ran 10.2 million hours in 2014
    - Total expected AWS usage for this test: 2.1 million hours (100x 2014 test)

- Tests are continuously being increased in scale
  - Explore limits of elasticity and overcome them
AWS Spot Price Market

▪ To be able to use the spot market efficiently, applications need to be “preempt-able” in one way or another
  ◦ spot price market instances are being shut down within X minutes when market price goes above bid ➜ goal is to minimize loss of work and maximize efficiency

▪ Solutions are being worked on
  ◦ Simplest solution is to shorten the processing time of jobs, accept efficiency losses and resubmit jobs
  ◦ Atlas accelerated their work on the Atlas Event Service:
    • This service permits a pilot job to perform units of work smaller than a full ATLAS job, e.g. about 10 minutes.
      - Intermediate results are stored in an object store.
      - These are later merged to create final output (identical to what would have resulted from a full-length job).
      - Intermediate objects can be discarded.

John Hover at HEPiX Spring 2015:
"Running ATLAS at scale on Amazon"
Evolution of the Grid site
Virtual Facility

- Not only experiments can benefit from the elasticity promise of commercial clouds

- Grid sites are starting to rethink their current setup
  - Instead of only static allocations, provide ability to dynamically expand resources depending on resource needs of users
  - Provide needed resources for users without provisioning owned resources for peak
    - Optimize balance between owned resources and “rented” resources
  - This will be an intense area of R&D in the near-term future

- Sites could start providing “complete solutions” for their users
  - CPU capacity with guaranteed level of service
    - Users would not have to care about whether their jobs are running on “owned” or “rented” resources
    - Sites could make the economic decision themselves and optimize their cost structure
  - Storage services that adapt to where the jobs are running
  - On-demand auto-scaling services
Transparent access for the Science Community
Open Science Grid

- Created out of the goal to share the LHC experiments’ Grid infrastructures and other Experiment/University/Lab infrastructures in the US amongst all HEP sciences and beyond.

- Major clusters at Universities & National Labs connected to the Grid:
  - Sharing policy is locally controlled.
  - All owners want to share to maximize the benefit to all.

- Researcher use a single interface to use resources …
  - ... they own
  - ... others are willing to share
  - ... they have an allocation on
  - ... they buy from a commercial (cloud) provider

- OSG focuses on making this technically possible for Distributed High Throughput Computing:
  - Operate a shared Production Infrastructure ➜ Open Facility (glideinWMS)
  - Advance a shared Software Infrastructure ➜ Open Software Stack
  - Spread knowledge across Researchers, IT professionals & Software developers ➜ Open Ecosystem

from: OSG All-Hands Meeting March 2015: Executive Director’s Update
How the Open Science Grid is used

- **Single PI Perspective**
  - **OSG-Connect:**
    - OSG operates login node, disk space, application software repo, and provisions resources across the facility for single PIs and small groups
  - **OSG-XD**
    - ~same functionality, but users are being redirected to OSG from HPC allocation committees (XSEDE)

- **IT Organization Perspective**
  - Universities/Labs use OSG technologies to “flock” local work to the OSG

- **Large Scale Research Community Perspective**
  - LHC experiments and other large VOs use the OSG directly

From: OSG Annual Report to DOE & NSF (March 2015)
Summary & Outlook

▪ Resource landscape for HEP is changing
  ◦ GRID is augmented by
    • Cloud
    • HPC

▪ Cloud
  ◦ Integration challenges are being worked on by many and it is exciting to see the progress
  ◦ What we should look out for:
    • When are the regular commercial Cloud resources becoming competitive?
    • How will we be able to benefit from the spot price markets?

▪ HPC
  ◦ Interesting solution for specific problems in HEP computing
    • Or maybe for more?

▪ Future of the Virtual Facility?

▪ How will other sciences continue to benefit from HEP’s large scale computing experience?
Questions

- Grid, Cloud and HPC have different resource allocation models
- The question:
  - How can we integrate these three different models?
  - Do we have to evolve the static allocation model we are used to?
Acknowledgements

▪ Many thanks for the invitation to the CHEP organizing committee.

▪ Big thanks to Gabriele Garzoglio for working with me on the talk while caring for his expanded family!

▪ Many thanks also to all my colleagues who helped preparing the talk and for their comments, especially:
  ◦ Stuart Fuess, Burt Holzman, John Hover, Bo Jayatilaka, Jim Kowalkowski, Ruth Pordes, Panagiotis Spentzouris, Steve Timm, Margaret Votava, Frank Würthwein