Clouds

IaaS like architecture tried at CERN and INFN, see presentation in first F2F

- Internal cloud approach
- Works with virtualization
- Optimized for speed rather than reliability
- Only trusted images
- CERN approach implements ideas and work of HEPiX virtualization working group
- Used for batch virtualization (small scale prototype)

Internal cloud approach, entirely transparent to the users

What for ?

- More dynamic batch (=unreliable) resources
  - All the good stuff of virtualization
  - Dynamic adaption of worker node mixture to requests
  - Automation of intrusive updates for worker nodes
  - Stuff like whole node provisioning is relatively easy
- Cheap opening to new computing models
  - Cloud controllers often come with an EC2 interface

Drawbacks

- Current cloud controllers have fairly basic schedulers
- No concept of infinite resources within WLCG
  - Not considered to be a problem
  - Need to have finite live time of VMs to get resources back
- Billing is currently missing in WLCG

Hybrid or Public clouds

- “public” = public to some VOs
- Policies, check HEPiX virtualization WG results
  - No user provided images
  - Need chain of trust for images
  - No run time root access for users
- Policies are site-dependent
  - Sites are free to relax policies
  - In that case security and traceability has to be ensured
  - Vlans, cgroups and containers may help here

Chances

- Model:
  - Sites give trusted worker nodes to VOs within quotas
  - Instances may directly call back to the experiment framework to get work load done (SaaS)
  - Those images could even be simple batch clients for the VO
- Tested based on CERNVM images by ATLAS and LHCb at CERN
- Tested at INFN with Auger

Stuff to be thought about ...

- Images and trust
- Characteristics of default images (similar to tags)
- Cloud access method (ec2 or occi or something else ?)
- fair share in this model requires some thinking. Some experiences with caching at INFN on this
- Life time of VMs can be long but not infinite
- Management of VMs after instantiation

Accounting and billing

- Do we need to change accounting ?
  - Wall clock time accounting with or without normalization?
- Do we need to introduce billing ?
  - Different “Quality of Service” domains, eg live migration support, local disks, reliable hardware, ...
  - Different VM sizes
  - Live time limitations for virtual machines, you pay for what you get

Recommendations?

- Sites are free to use virtualization. If they do, it must be transparent to users
- Sites which decide to go beyond virtualization, and setup an internal cloud, are free to open part of the resources via an EC2 access to VOs which are interested in this. These resource do not come for free. The total resources (batch + cloud) sums up to the pledges for the VO.
- Free protocols for Cloud access are to be preferred over proprietary formats like EC2. Realistically though, for the time being EC2 seems to be the most popular access method.
- virtual machine sizes definitions should be unique across all sides, eg m1_small = 1cpu, 2GB memory.
- In the described IaaS models, experiments need to integrate their frameworks with clouds. We recommend to found a body which involves all parties to synchronize these efforts, with the aim to avoid independent developments and multiple solutions for the same problems.
- Virtual machine live times have to be limited to ensure a dynamic system