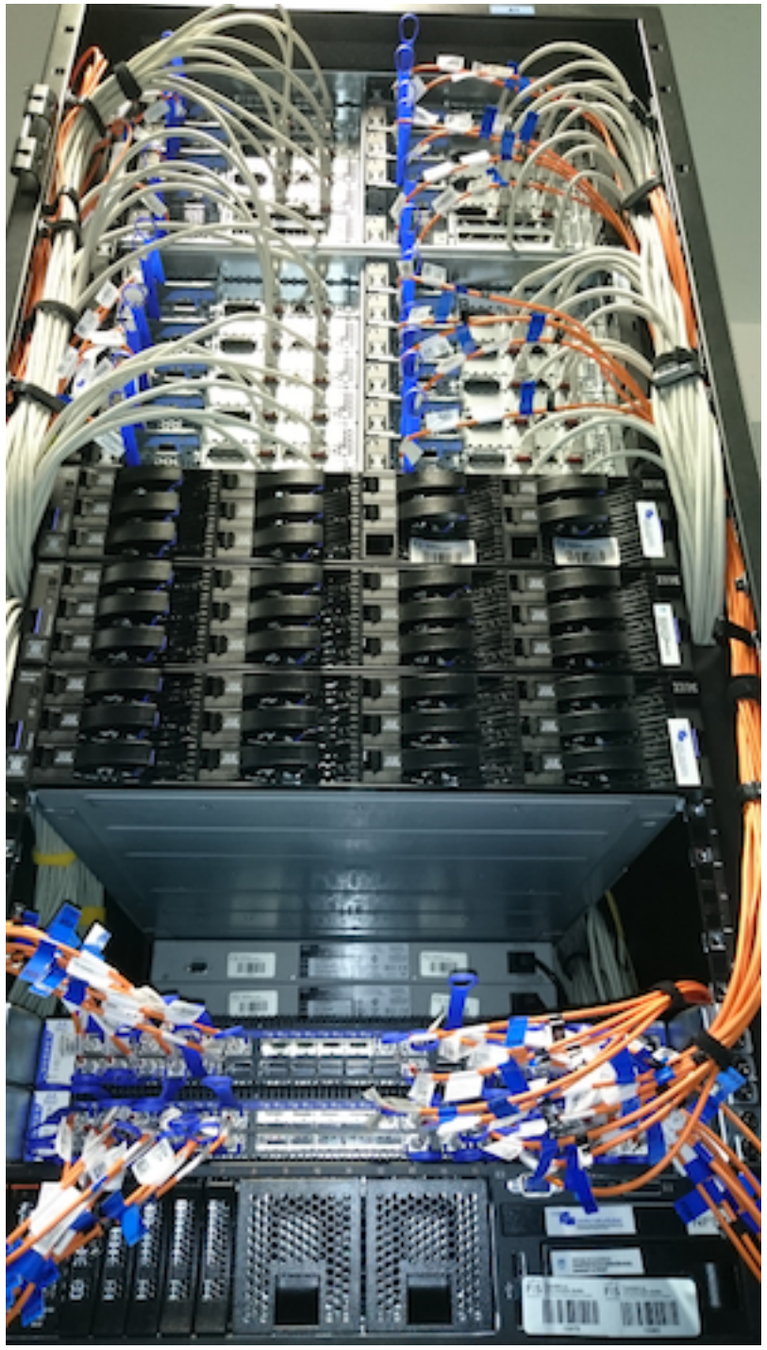
Distributed Computing as a Service with ARC

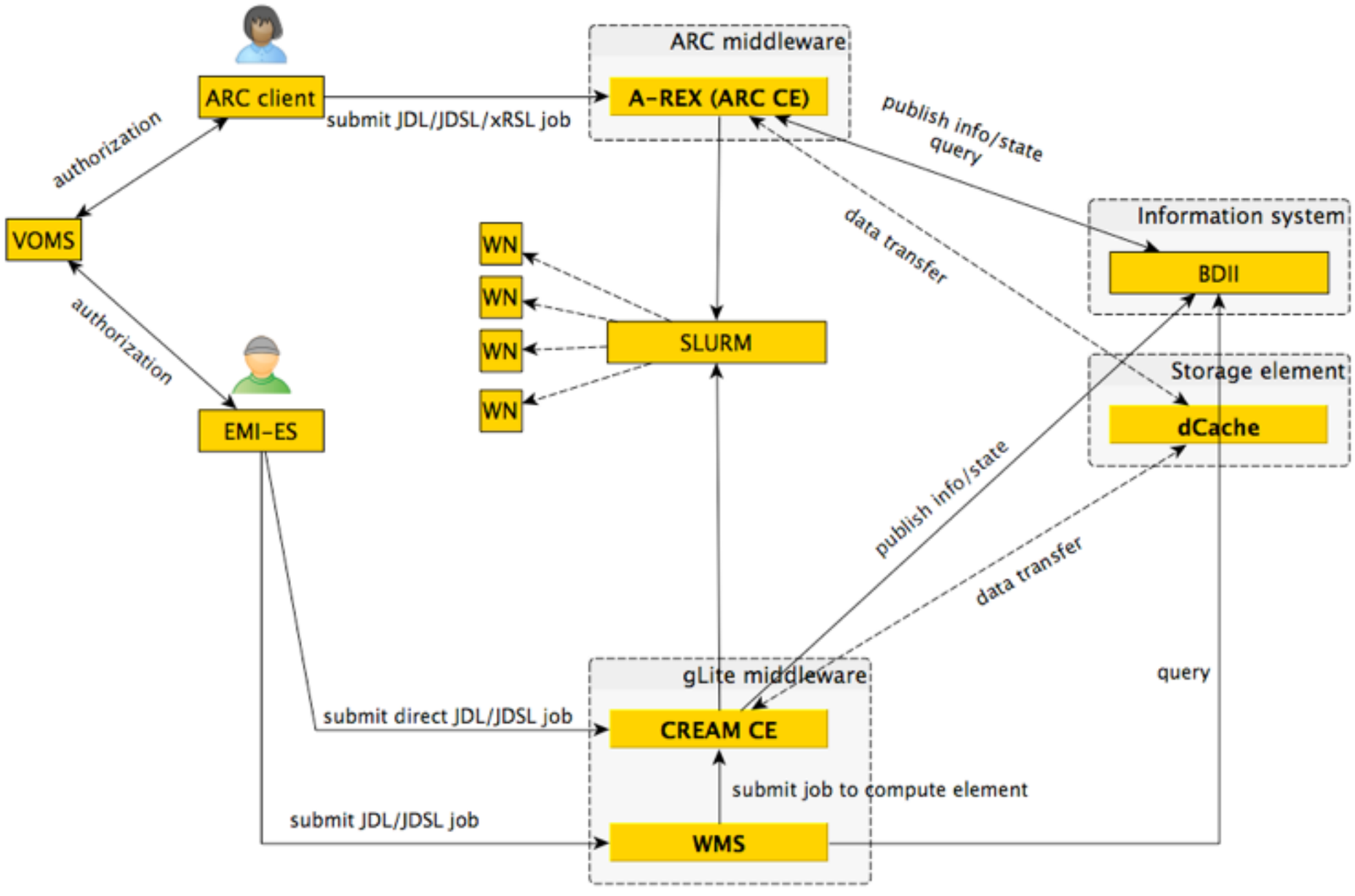
Barbara Krašovec

Arnes

- Slovenian NREN, Member of Sling (Slovenian NGI, part of EGI)
- Infrastructure:
  - HPC cluster,
    grid cluster (~4000 cores)
    - ARC, gLite middleware
    - SLURM batch system
  - OpenStack cloud testbed (~100 cores)



Current grid infrastructure



Motivation

- Customization of execution environments (flexibility).
- Additional resources in peak times - cloud bursting (scalability).
- Extension of grid resources with cloud resources without any modifications on users' side

Integration difficulties

- VM provisioning (vCycle, Cloudscheduler, Libcloud, Terraform, condor_rooster)
- authz and authn (VOMS plugin),
- data management (data intensive tasks),
- instance types (proprietary image formats and resource sizes),
- registration of resources (infosys),
- nonstandardized API-s,
- performance penalty.

Integration benefits

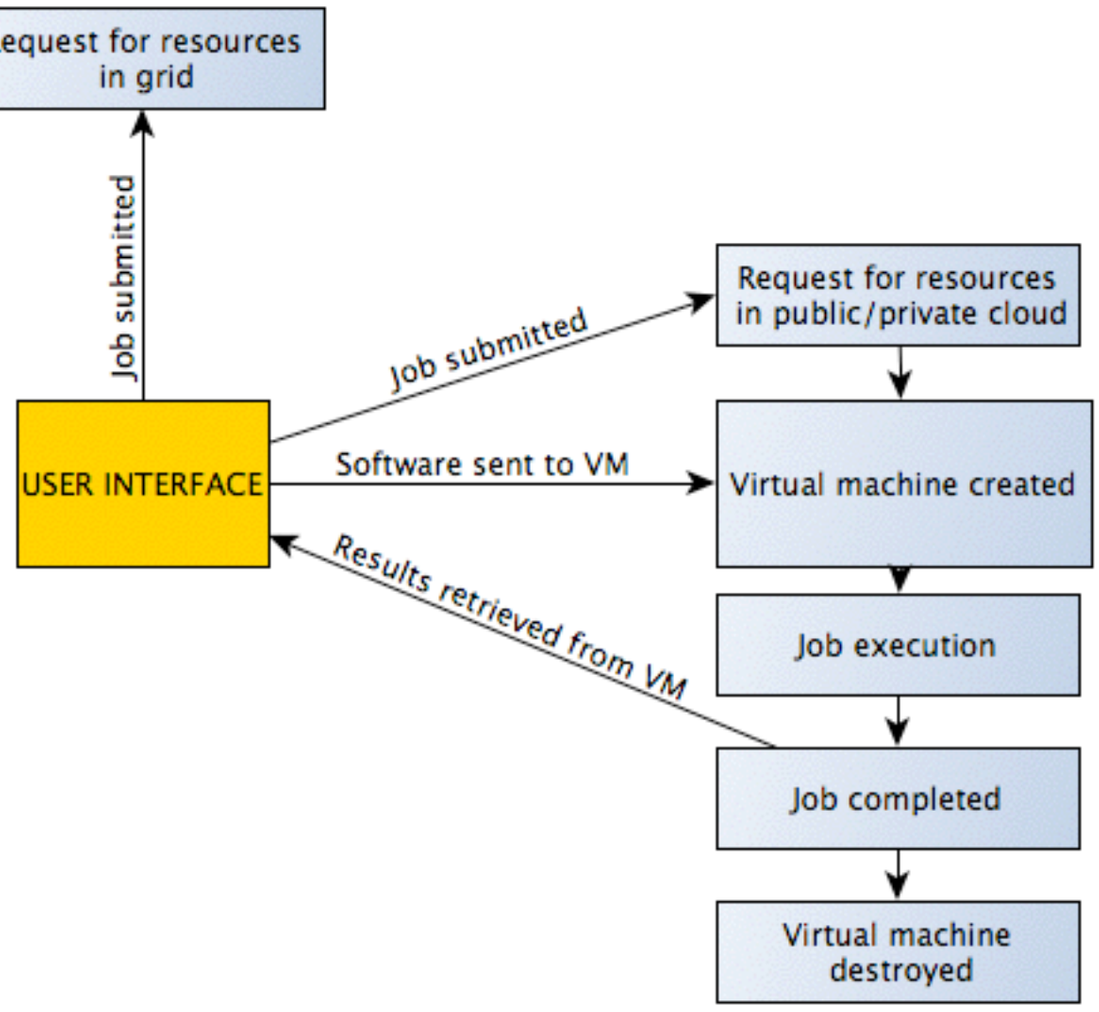
- portability of applications,
- optimal resource utilization,
- customized execution environments,
- cloud bursting,
- high availability and reliability,
- scalability.

Integration models

- Unified access to grid and cloud.
- Grid and private cloud integration.
- Grid virtualization.
- Grid and public cloud integration.

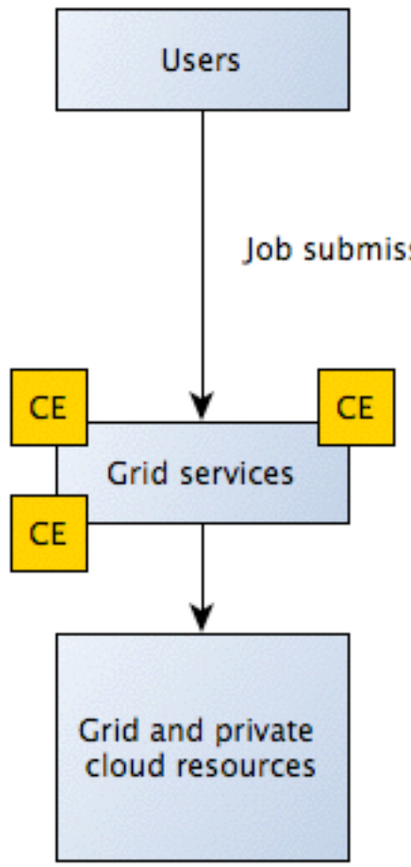
Unified access to grid and cloud

- Common interface to submit to different infrastructures
- Solutions:
  - DIRAC, EMI-ES, Swarm, XtremWeb, HTCondor client



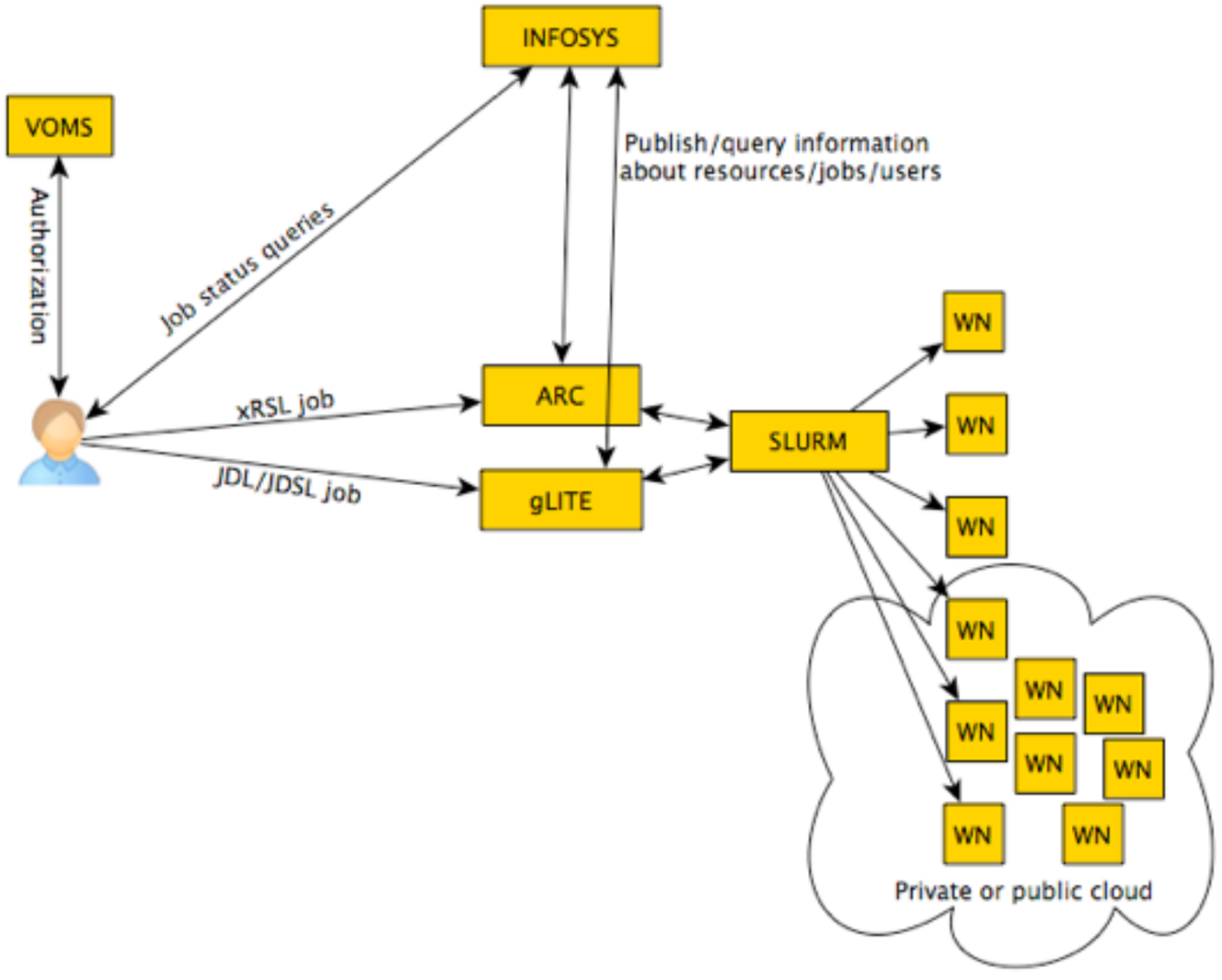
Grid and private cloud integration

- Solutions:
  - WNoDeS,
  - ARC Rainbow.
- Goal: hybrid LRMS and VM-aware ARC cluster



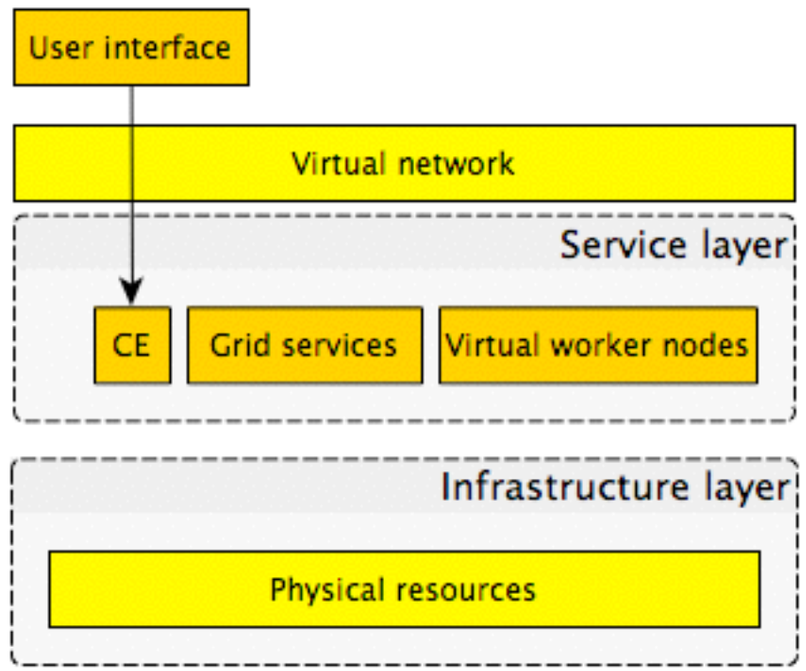
Grid and public/hybrid cloud integration

- Extending grid cluster to the hybrid cloud
- Interoperable LRMS
- Similar implementation at University of Goettingen: integration of Unicore grid cluser, Eucalyptus and Amazon cloud



Cloud federation - grid virtualization

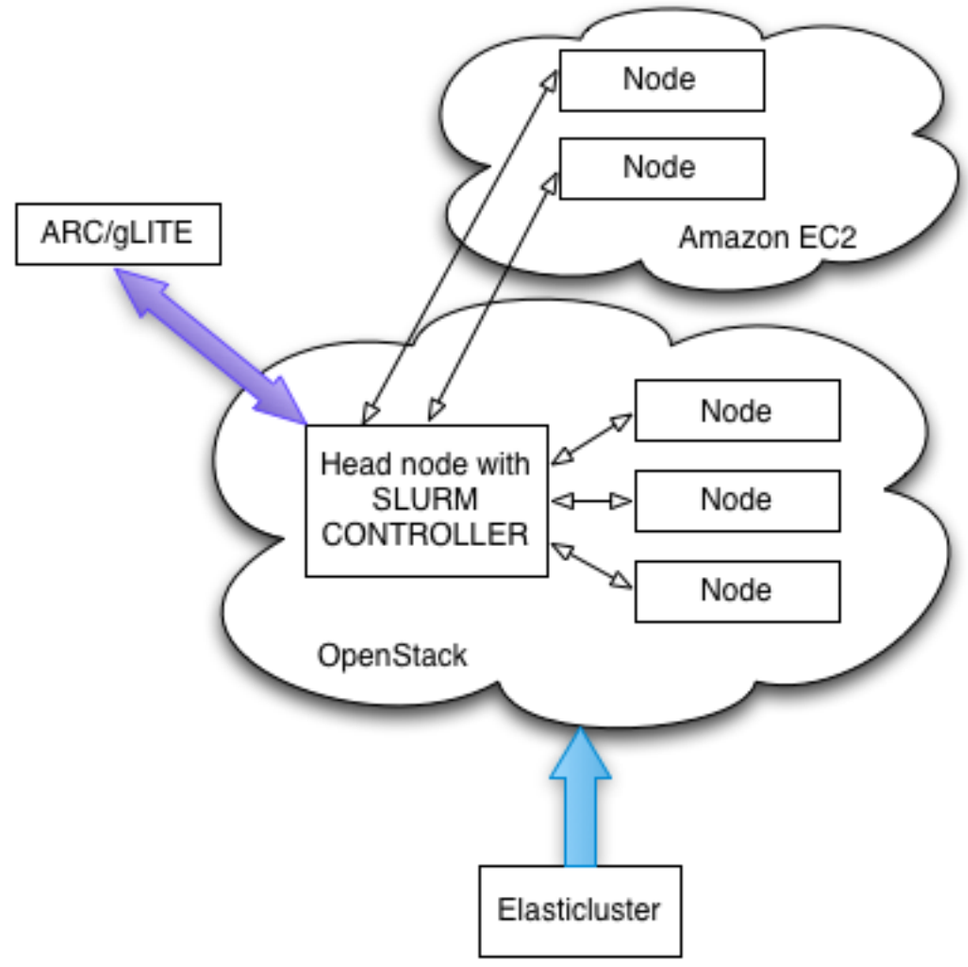
- Multiple academic clouds:
  StratusLab
  Federated Cloud
  Venus-C
  Aneka Tools
  Helix Nebula
  Nimbus



Integration possibilities (1)

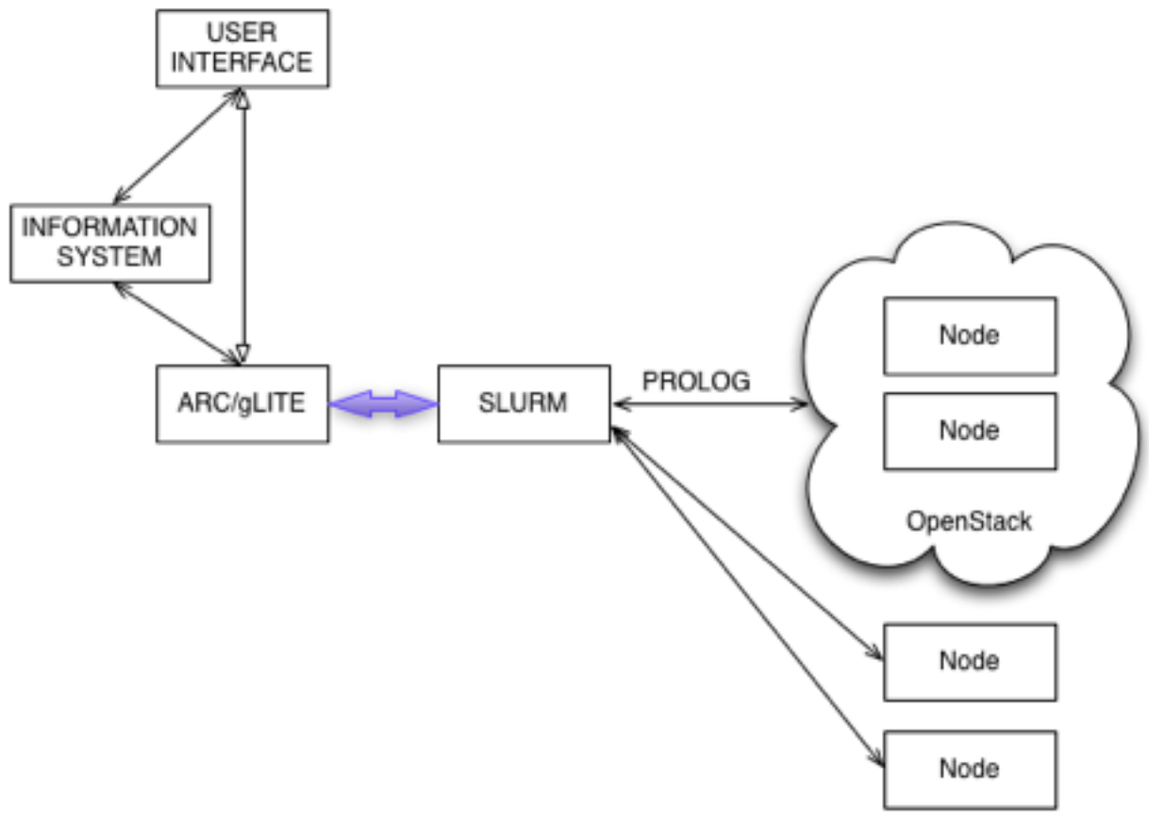
- Elasticluster:
  cluster of VMs
  on EC2/OpenStac
  + SGE/PBS/SLUR

- Similar:
  STARCluster:
  cluster of VMs
  on EC2



Integration possibilities (2)

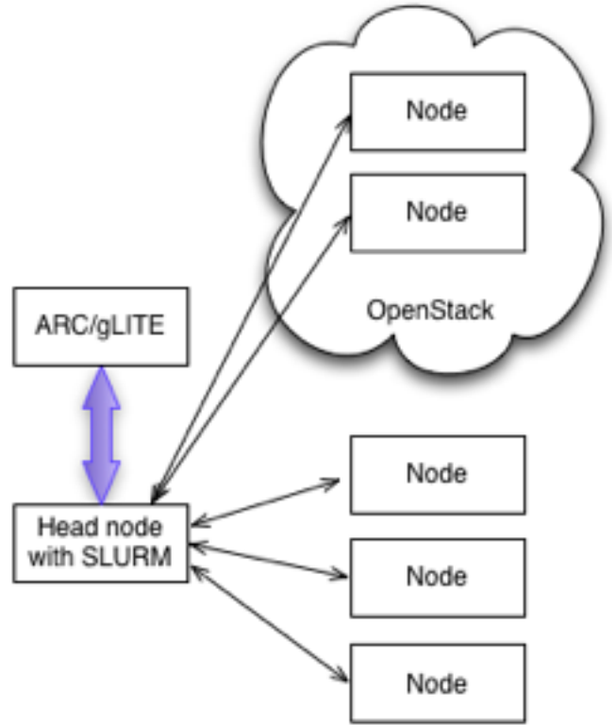
- ARC + SLURM
  prolog script
  + OpenStack



- limited to a single cloud provider
- when public cloud, implementation of billing required

Integration possibilities (3)

- ARC + SLURM
  cloud plugin + Amazon EC2
  (Slurm Elastic Computing)



- Suspend/ResumeProgram
  SLURM_NODE_ALIASES env requir
  New cloud partition

```
NodeName=cloud[001-100] Weight=8 Feature=cloud State=cloud
```

  Add IP-address and hostname

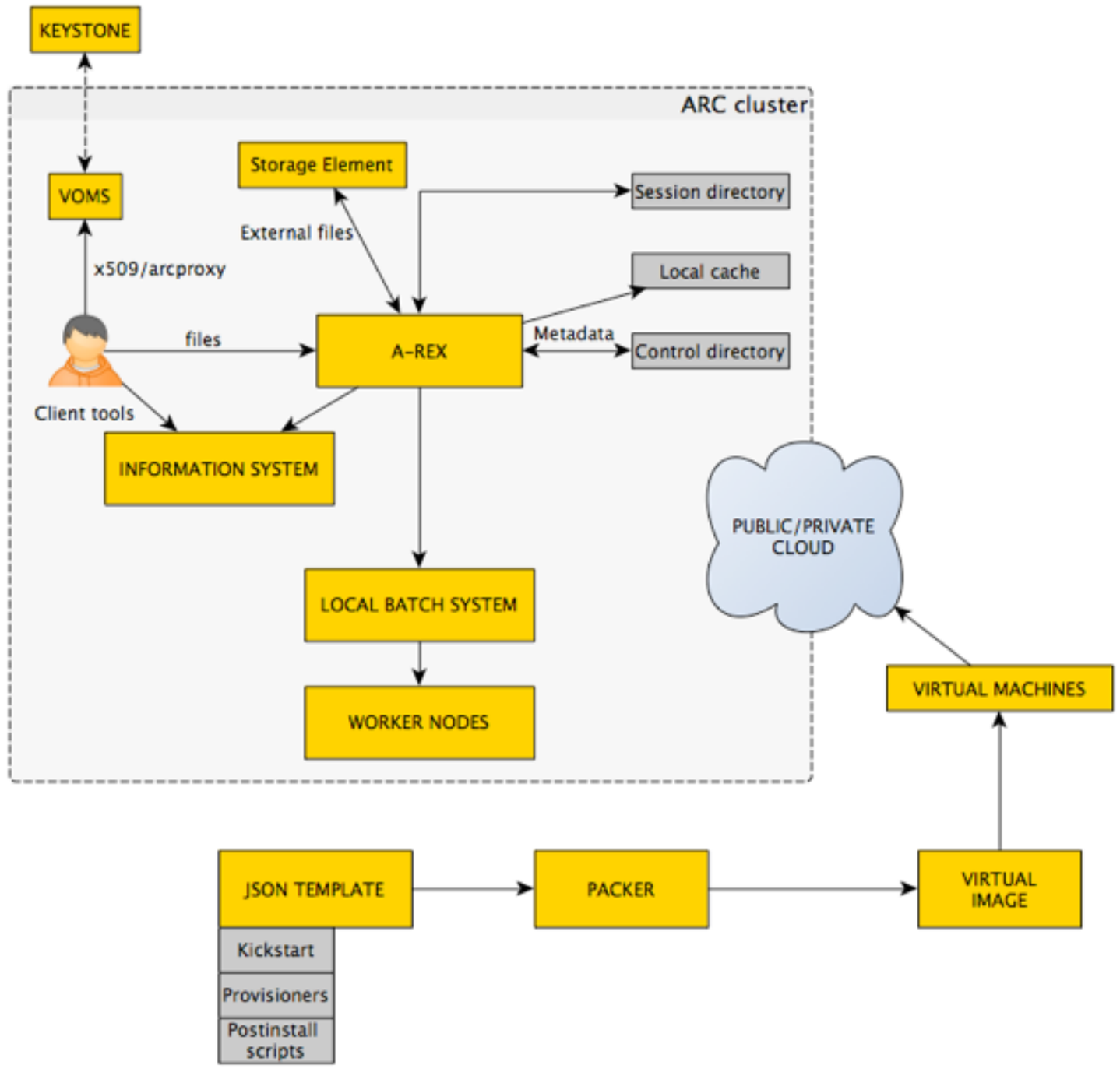
```
scontrol update nodename=cloud001 nodeaddr=<IP
>nodehostname=cloudwn
```

- Some manual work required each time

Integration possibilities (4)

- Other possibilities:
  - Interoperable ARC client to submit jobs to grid and public cloud(s)
  - Cloudscheduler + HTCondor + ARC + OpenStack -> Arc cluster virtualization

ARC Cluster in the Cloud (ARC-CC)



ARC-CC basic setup

- VM provisioning done manually via API/ AWS/Horizon.. Security rules pre-defined.
- Built with Packer and post-install scripts.
- Headnode installed, then worker nodes added.
- RTE-s included in the VM image,

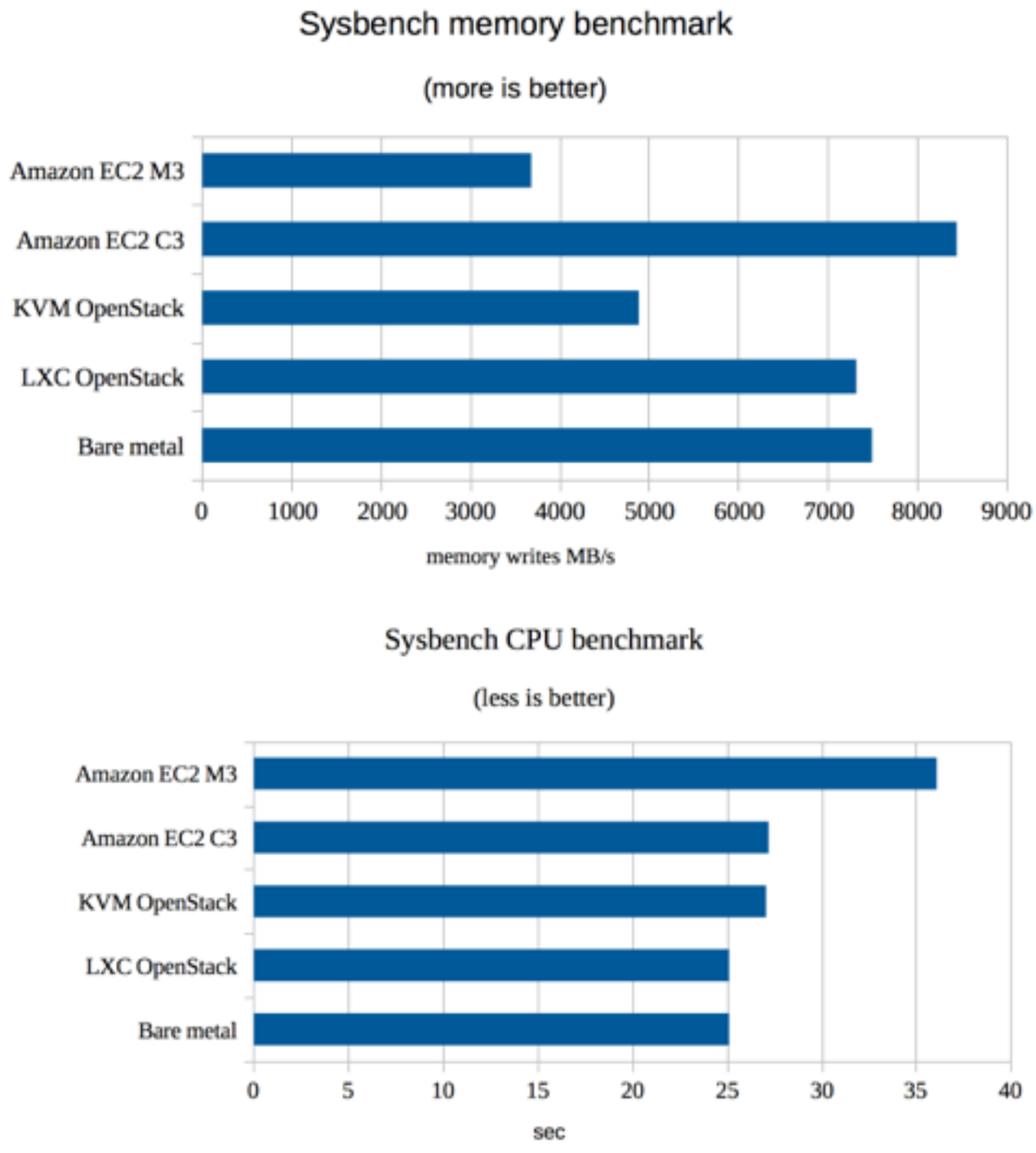
ARC-CC authz and authn

- OpenStack: VOMS Keystone plugin https://github.com/IFCA/keystone-voms
    - problem with expired credentials
- Amazon EC2: self-signed CA or EUgridPMA certificate + VOMS support (post-install script)

ARC-CC performance evaluation

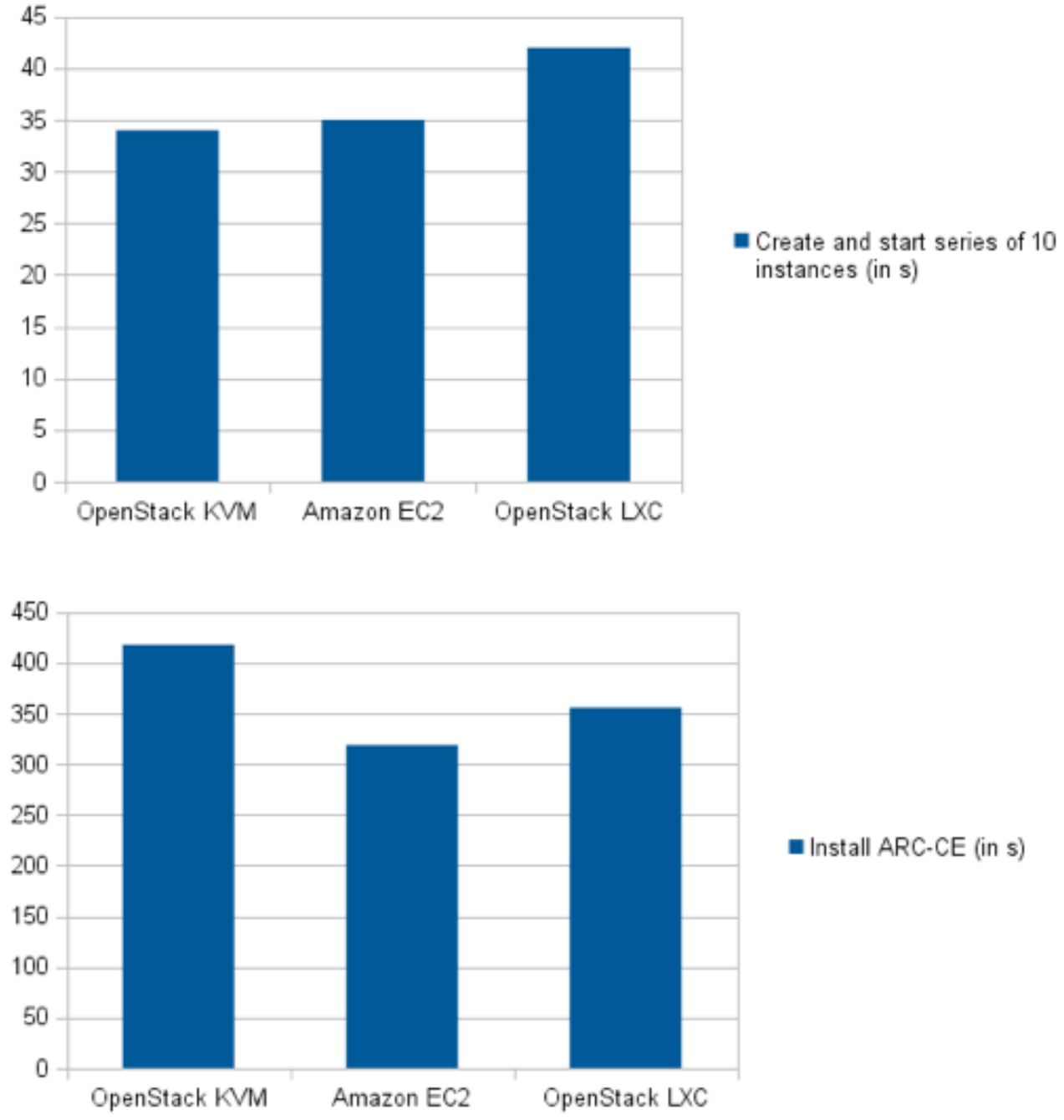
- Tested on compute instances of Amazon EC2 (instance c4-xlarge Xeon E5-2680 v2 Ivy Bridge, 4 CPU, 7.5GB RAM, SSD storage) and OpenStack cloud (Xeon E5-2650 v2 processor and same resources sizes)
- ARC-CC compared to bare metal:
  - Memory overhead only 1%
  - CPU overhead 7-8%

CPU and memory (Sysbench)



Building ARC-CC in the cloud

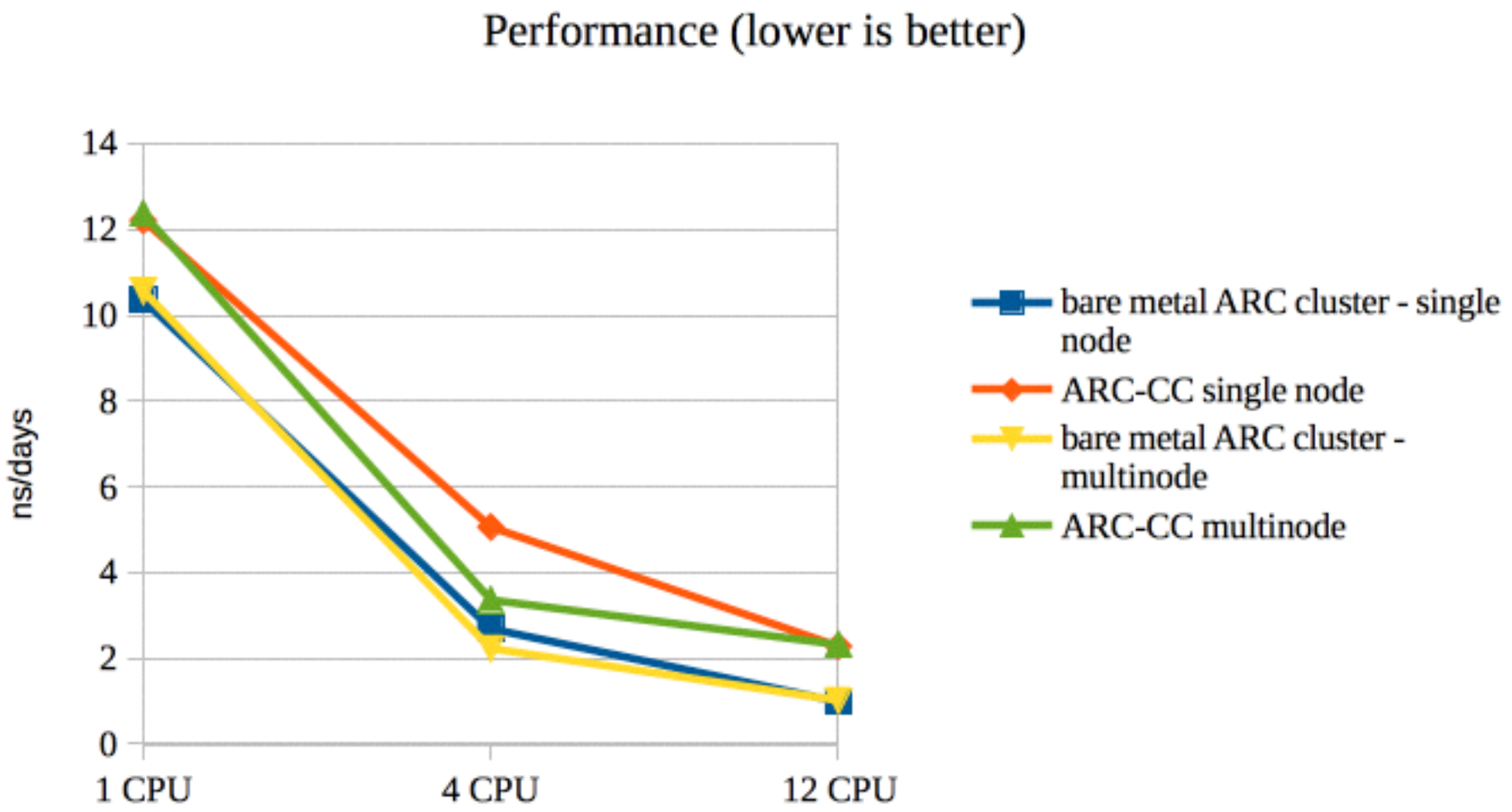
- VM provisioning is fast
- virtual ARC-CE built within 5-6 minutes



NAMD simulations

- NAMD - scalable molecular dynamics (using apoa1 benchmark, 500 steps of simulation)
- testing scalability over 4 nodes
- inconsistent results (up to 15% performance differencies) - warm-up phase, noisy neighbors?
- overall performance 15% lower on ARC-CC compared to bare metal ARC cluster
- scalability efficient in all environments

Overall NAMD performance



- Days of computation time required per nanosecond of simulation

Conclusion

- Simulations on ARC-CC: 15% performance penalty compared to a physical cluster. Almost no memory overhead. Some high latency problems.
- Data-intensive tasks should be tested.
- Performance depends also on noisy neighbors, type of instance etc.
- Usable solution for peak times and legacy code execution.
- Work to be done: VM images for VO-s, VM provisioning

- Thanks. Questions?